The Thesis Committee for Mohammed Zaki AlQassab
Certifies that this is the approved version of the following Thesis:

Modeling Hydraulic Fractures Using Microseismic Events

APPROVED BY
SUPERVISING COMMITTEE:

Kamy Sepehrnoori, Supervisor

Wei Yu

Modeling Hydraulic Fractures Using Microseismic Events

by

Mohammed Zaki AlQassab

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

August 2020

Dedication

To my wife, **Abrar**, and my two-month old son, **Hashim**

Acknowledgements

First, I would like to thank Dr. Kamy Sepehrnoori for his teaching, guidance, and support. I used several numerical methods in this thesis, and I learned those methods from Dr. Sepehrnoori's courses. Also, I would like to thank Dr. Wei Yu for closely following up with my research progress and providing valuable ideas and recommendations that improved my thesis.

In addition, I would like to thank Weitong Sun for sharing with me his workflow and code for assisted history matching. He helped me start my research and inspired me to improve my coding skills.

I would like to thank the companies CNPC and SM Energy for providing the field data. I also appreciate the discussion with Erich Kerr from SM Energy and his suggestions to improve our workflow.

Finally, I am grateful to Saudi Aramco for sponsoring my graduate education.

Abstract

Modeling Hydraulic Fractures Using Microseismic Events

Mohammed Zaki AlQassab, M.S.E.

The University of Texas at Austin, 2020

Supervisor: Kamy Sepehrnoori

Advancements in hydraulic fracturing technology have enabled the development of unconventional reservoirs. Hydraulic fractures increase the total surface area of the wellbore, which leads to an increase in production rate. One way to evaluate the success of hydraulic fracturing jobs is to detect microseismic events during fracturing. Mapping microseismic events help engineers identify the areal extent of the fractures. However, estimating the actual size, shape, and orientation of hydraulic fractures from microseismic events is challenging because microseismic events are week signals and include noise (Warpinski 2009).

Here we propose a novel workflow that builds a discrete fracture model directly from microseismic events. We use several techniques such as density-based spatial clustering of applications with noise (DBSCAN), surface fitting, embedded discrete fracture model (EDFM), and proxy-based assisted history matching (AHM).

We first define the region for each stage using the perforation intervals. Then, we use DBSCAN to reduce noise and identify clusters in each stage. Next, we choose the main

cluster in each stage to fit a fracture plane to the microseismic events. The last step is to calibrate the fracture model using two scaling factors: one reduces the fracture height and the other reduces the fracture half-length. We determine the appropriate scaling factors using AHM. Therefore, the final calibrated fracture model would match field production data. We found that preliminary fracture model overestimates the size of the fractures. Hence, calibrating the fracture model with production data is important.

There are several field applications that can benefit from our workflow. For example, we can compare the fracture models for several offset wells in a reservoir and make some correlations with their fracturing strategies. The best fracturing strategy can then be implemented for future wells.

We also introduce a new approach that estimates bottom hole pressure from static wellhead pressure in wellbores filled with gas and water. We divide the gas column into (n) small segments. Then, we evaluate the pressure in each segment along with the depth of the gas-water interface by numerically solving (n+1) equations. This approach is useful in history matching since obtaining bottom hole pressure is challenging and expensive.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Reservoir permeability is a key factor that influences the production rate in oil and gas wells, as suggested by the famous Darcy's law. Permeabilities in the unconventional reservoirs are extremely low compared to conventional sandstone and carbonate reservoirs. Typical permeabilities in the unconventional reservoirs are in the micro and nano Darcy scales. Hence, production in the unconventional resources are low and often uneconomical. In fact, engineers did not consider shale formations as oil/gas producing reservoirs for a long time. However, as hydraulic fracturing technology advances, producing oil and gas economically from unconventional reservoirs became possible. The main advantage of hydraulic fracturing is increasing the wellbore surface area which leads to an increase in production rate.

During the hydraulic fracturing process, companies inject a large volume of fracturing fluid into the wellbore at high pressure that results in shear failures in the formation. Companies can detect the shear failure positions during a hydraulic fracturing job by placing geophones in an offset well, **Figure 1.1**. The collected data are known as microseismic events. Plotting the location of the microseismic events helps us understand the extend of the created hydraulic fractures.

Companies spend a lot of time and money to build well-calibrated reservoir simulation models. The models are then used to understand the flow mechanisms in the reservoir, forecast production rates and develop future business plans. Therefore, the accuracy of the reservoir models is important. Incorrect models could cause companies to lose money. However, including accurate hydraulic fracture properties in reservoir simulation models is challenging. After fracturing a well, there is no direct method to measure the fractures' properties such as fracture half-length, fracture height, azimuth, and

dip angle. Engineers can estimate the properties based on near-wellbore measurements, well production, or pressure behavior. However, these methods are unable to predict actual fracture dimension and orientation (Warpinski et al. 2013). A direct diagnostic technology such as microseismic events has the potential to provide a better estimate for hydraulic fracture properties.

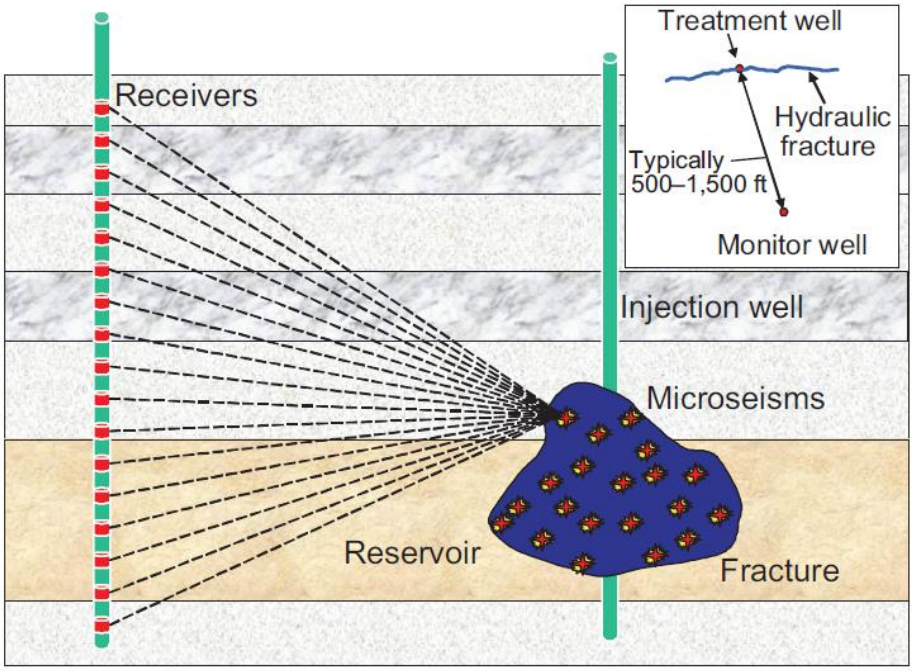

Figure 1.1: Schematic of microseismic monitoring. Geophones are placed in an offset well to detect the microseismic events around the treatment well (Warpinski 2009).

Another challenge, in modeling fractured wells, is representing the hydraulic fractures in a reservoir simulation model. To include fractures directly in a reservoir model, we need to use an extremely fine grid model. Ideally, the grid block size should be the same or smaller than the fracture aperture. Unfortunately, using a fine grid in reservoir simulation is not practical because it would take a long time to run a simple model; especially if we

are trying to perform sensitivity or history matching studies. Therefore, researchers have developed other methods to represent or include the effect of fractures in reservoir simulation models. One efficient and easy-to-implement method is called embedded discrete fracturing modeling (EDFM). EDFM uses two different domains: one for the matrix blocks and another for the fractures' blocks. The connection between the two domains are treated as source or sink terms in the material balance equation (Li and Lee 2008).

Building a well-calibrated reservoir simulation model requires a lot of time and effort. Specifically, history matching is the most time-consuming step in the process. History matching problems are not linear. It is possible to match production data using different solutions. In addition, there is a large degree of uncertainty in the collected data from unconventional reservoirs because of their heterogeneity. In addition, performing lab tests such as permeability measurements for unconventional reservoirs is more complicated compared to conventional reservoirs. Hence, we expect the data to be inaccurate and we might not be able to reproduce the result using a different method or even a different lab. Therefore, we must not rely on only one history matching solution. The decision maker should consider a range of solutions to make sound judgment in the field. One method to overcome the history matching challenge is to use assisted history matching (AHM). Instead of requesting the simulator to run every possible scenario or sample, AHM helps us select the appropriate samples to be tested next.

1.1 RESEARCH OBJECTIVES

The objective of this research is to establish a direct method that builds a discrete fracture model using microseismic events. The fracture model should honor the distribution of the microseismic events and be able to match the production data from the field. The

created hydraulic fractures in the field are not uniform. Each fracture has its unique properties. However, in most reservoir simulation models, engineers assign the same fracture half-length, height, azimuth, and dip angle to all the fractures in a single well. Microseismic events have the potential to identify the unique properties for each fracture.

1.2 THESIS OUTLINE

This thesis consists of seven chapters and the following is a brief describing of each chapter.

Chapter 2 is a literature review of modeling hydraulic fractures using microseismic events. Also, we review the different methods used to include the effect of fractures in reservoir simulation models.

In Chapter 3, we introduce our workflow and explain how we perform each step. We also include some general examples for illustrations.

Chapter 4 demonstrates a novel technique that converts wellhead pressure to bottom hole pressure in a wellbore that has both gas and water. This topic was not the main interest of this research. However, it was needed to improve the accuracy of our model in one of our case studies.

Chapter 5 and Chapter 6 present case studies using our workflow in a gas condensate reservoir and a shale gas reservoir, respectively.

Finally, Chapter 7 summarizes the thesis and includes some recommendations for future studies.

Chapter 2: Literature Review

2.1 Fracture Modeling Using Microseismic Events

Building a fracture model from microseismic events is challenging because microseismic events include a lot of noise. Microseismic events are weak signals and can be affected by other activities around the wellbore. Noise could be the result of some fluid flow in the wellbore or drilling activity/well intervention in an offset well (Warpinski 2009). In addition, microseismic events from different stages could overlap, **Figure 2.1**. A possible explanation is that shear failures continue to occur even after moving to the next stages. Liu et al. (2017) proposes a method to reduce the stage overlapping issue. They divide the microseismic events for each stage into three windows: the pad window, the proppant window, and the closure window. They rely on the recoded time for each microseismic event to select the appropriate window. Then, they disregard both the pad and proppant windows and only use the closure window.

There are two main approaches to build a fracture model from microseismic events. The first approach is to use the geomechanical properties of the reservoir rock and the pumping information to estimate the fracture properties (Xu et al. 2009). Then, the uncertain parameters in the model can be adjusted to match the microseismic events. In this approach, we can obtain additional information such fracture width and fracture permeability. However, one disadvantage of this approach is that the result is theoretical and not based on direct measurements.

The second approach is to match the microseismic events directly. For example, Yu et al. (2016) uses moment-tensor analysis and a Hough-transform method to build a fracture model from microseismic events. The method we are presenting in this thesis follows the second approach.

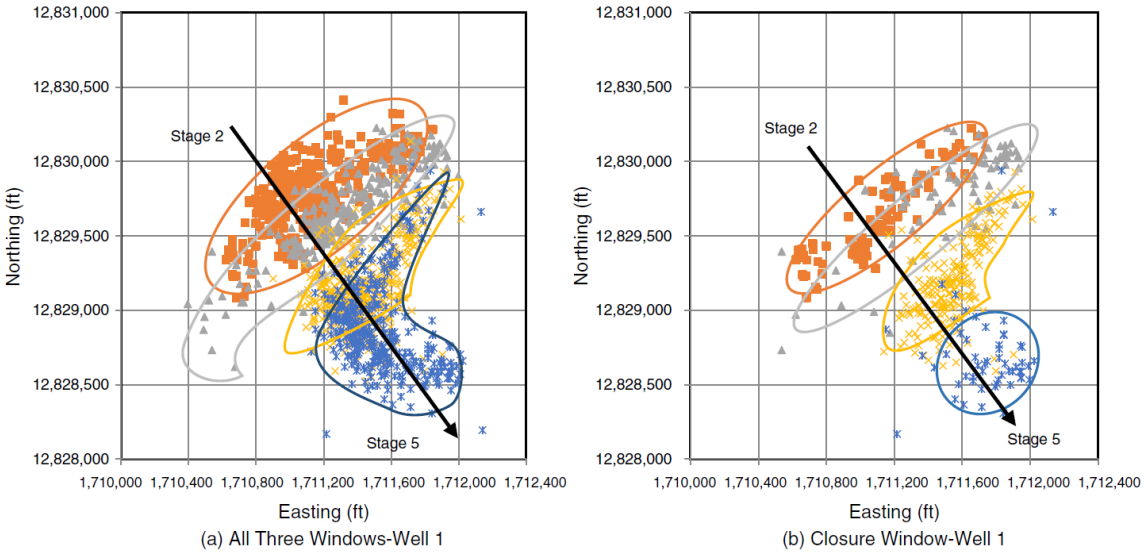

Figure 2.1: Overlapping between different stages. (a) original microseismic data. (b) microseismic data with less overlapping after implementing the method described in Liu et al. 2017.

2.2 REPRESENTING FRACTURES IN RESERVOIR SIMULATION MODELS

Researchers have developed several methods to represent or include the effect of fractures in reservoir simulation models. Here we review the development of four approaches.

2.2.1 Dual porosity model

Warren and Root (1963) developed an analytical solution to model naturally fractured reservoirs. They considered a system with two regions: the matrix region with large pore volume but its flow capacity is negligible, and the fracture region has a small pore volume with large flow capacity. Based on this system, porosity can be classified into two types: primary porosity in the matrix region and secondary porosity in the fracture region. To simplify the problem, Warren and Root assumed the following:

- The matrix region is homogeneous and isotropic.

- The matrix can be represented as identical, rectangular parallelepipeds and the reservoir can be idealized as in **Figure 2.2**.
- The fractures are arranged in an orthogonal system. They are also uniform, continuous, and spaced equally.
- Fluid flow is not allowed within the matrix region. Fluid can flow in the fractures and between the matrix region and the fractures.

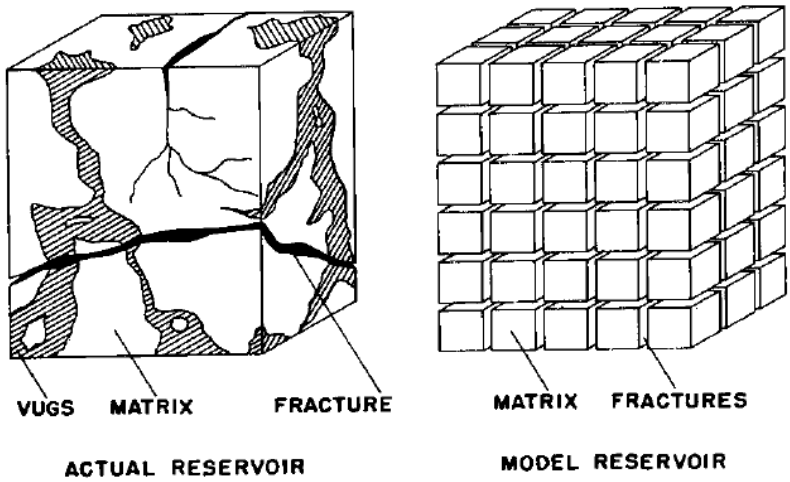

Figure 2.2: Reservoir idealization in the dual porosity model (Warren and Root 1963).

2.2.2 Dual permeability model

The dual permeability approach is an extension of the dual porosity approach. The main difference between the two approaches is that fluid flow is allowed between the matrix blocks in the dual permeability approach. **Figure 2.3** shows a schematic of a one-dimensional flow using three simulation methods: single porosity, dual porosity, and dual permeability (Dean and Lo 1988). In the single porosity approach, there is no fractures in the system. In the dual porosity approach, there are fracture and matrix blocks, however, fluid flow happens only between the fracture blocks and the matrix blocks are treated as a

storage volume for the fluid. In the dual permeability approach, fluid flow can happen between the matrix blocks and between the fracture blocks. Also, fluid exchange between matrix and fracture blocks is allowed.

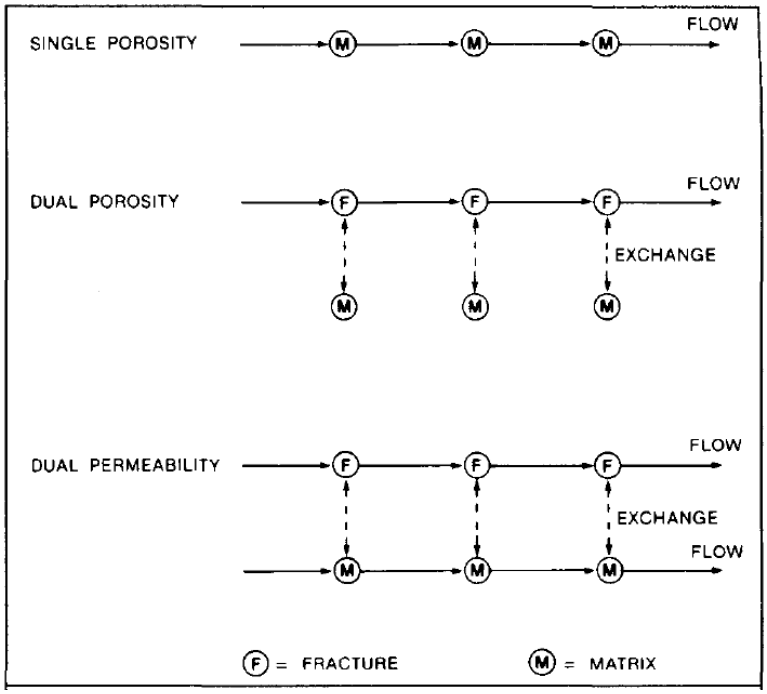

Figure 2.3: Comparison of one-dimensional flow using three simulation approaches: single porosity dual porosity, and dual permeability (Dean and Lo 1988).

Dean and Lo (1988) developed a three-dimensional, black oil reservoir simulator that uses the dual permeability approach. It can also be converted to the dual porosity approach by setting the matrix transmissibility to zero. **Figure 2.4** illustrates that each grid block may contain many matrix blocks. However, the matrix and fracture pressure and saturation are assumed to be constant within each grid block.

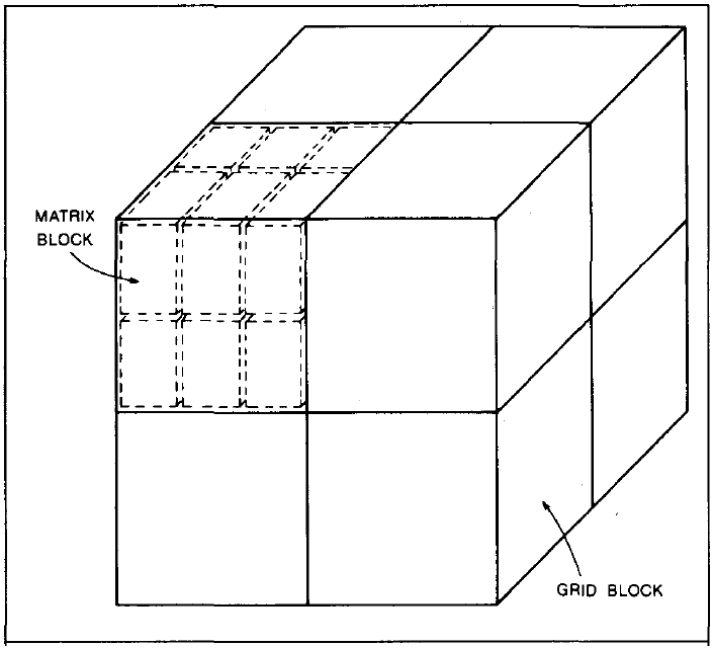

Figure 2.4: Grid blocks contain several matrix blocks in the dual permeability model (Dean and Lo 1988).

2.2.3 Discrete fracture model (DFM)

Both the dual porosity and the dual permeability approaches assume connected fracture network. For disconnected fractures, another approach was developed that is known as discrete fracture modeling (DFN). DFN represents fractures individually through techniques such as finite element or finite volume methods. Karimi-Fard et al. (2004) introduces a model that uses unstructured grid and finite volume method to represent a fractured reservoir. **Figure 2.5** illustrates how to convert a physical domain into a grid domain. For a 2D problem, polygons are used to represent the matrix and segments are used to represent the fractures. For a 3D problem, the matrix is represented by polyhedrons and the fractures are represented by polygons. The main advantage of the model is its compatibility with any reservoir simulator.

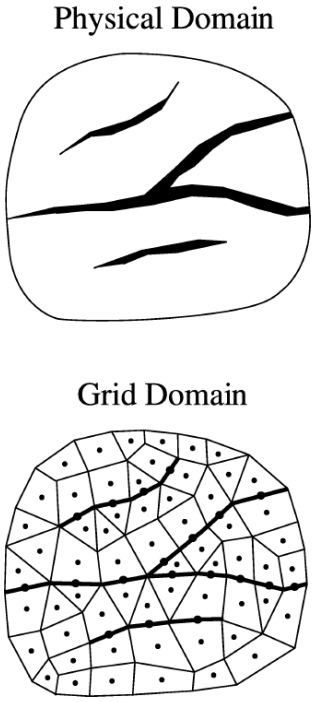

Figure 2.5: Discretization of the physical domain in the discrete fracture model (Karimi-Fard et al. 2004).

2.2.4 Embedded discrete fracture model (EDFM)

Li and Lee (2008) introduced a new approach, which is now known as embedded discreate fracture modeling (EDFM). They modeled vertical fractures explicitly and used two different domains: one for the matrix blocks and one for the fracture blocks. The interaction between the two domains is treated as a source/sink term in the material balance equation. The source/sink term can be expressed as the following

$$q_l = {q_l}^{wm} + {q_l}^{wf} + {q_l}^{mf}$$

where ${q_l}^{wm}, {q_l}^{wf}$ and ${q_l}^{mf}$ represent the flow rate of phase l between the well and the matrix, between the well and the fracture block, and between the matrix block and the fracture block, respectively. The first term (${q_l}^{wm}$) is already used in conventional reservoir

simulators. In addition, they used the concept of the transport index, which is similar to the well productivity index (PI), to calculate the second and third terms ($q_l{}^{wm}$ and $q_l{}^{mf}$).

Moinfar et al. (2014) expanded the basic 2D EDFM approach to the 3D EDFM and accounted for complex fractures with arbitrary dip angles. Since fractures intersect matrix blocks at different angles, the shape of the intersection is a polygon with three, four, five, or six vertices, **Figure 2.6**. The transport index is a function of the formed polygon area. The EDFM method is considered non-intrusive because it can be incorporated in most conventional reservoir simulators (Xu et al. 2017, 2019; Xu and Sepehrnoori 2019). **Figure 2.7** illustrates how the EDFM method works for 3D complex fractures. Three types of connections can be identified: between matrix and fracture, between fracture segments in a single fracture, and between intersecting fracture segments.

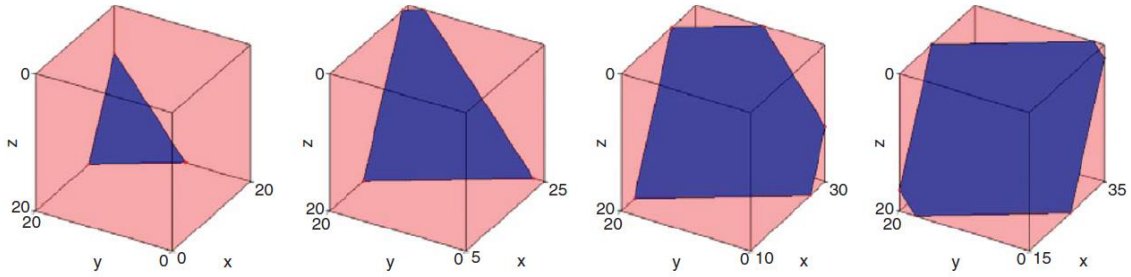

Figure 2.6: Possible polygons formed by the intersection of a fracture and a matrix block (Moinfar et al. 2014).

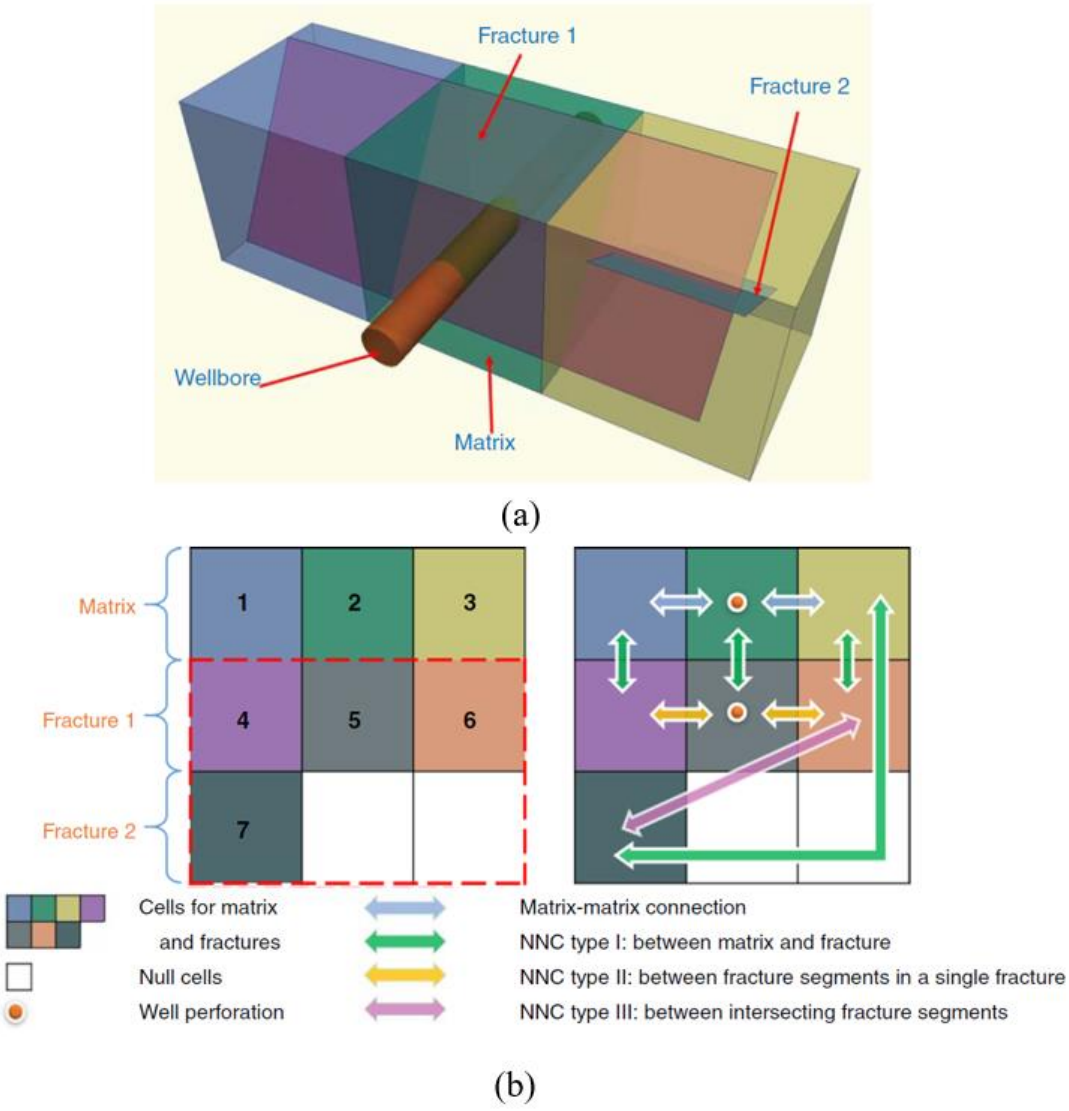

Figure 2.7: Illustration of the non-intrusive EDFM method for modeling 3D complex fractures (Xu et al. 2017): (a) Complex fractures in physical domain; (b) Matrix and fracture cells in computation domain.

Chapter 3: Workflow to Build Fracture Models using Microseismic Events

3.1 ASSUMPTIONS

In this chapter, we introduce our workflow that builds a discrete fracture model using microseismic events. We started with some general assumptions to simplify the problem and reduce the level of noise in the microseismic data. First, we assume that microseismic events in the vicinity of the wellbore are caused only by shear failure in the hydraulic fractures. We acknowledge that microseismic events can also be caused by activation of natural fractures around the wellbore (Maxwell et al. 2015). However, we made this assumption for simplicity and we recommend re-examining this assumption for future studies.

Second, we assume that all microseismic events are induced in the targeted formation. In the reality, geophones may detect microseismic events from shallower or deeper formations. Also, we represent fractures with planar features. Geologists and engineers generally accept representing fractures as planar features in reservoir models. Even for complex fractures, it is common to discretize a fracture into small planar features. Finally, to reduce the noise in the microseismic data, we assume that each fracture will generate a high-density cluster and those microseismic events that do not belong to clusters are outliers or noise.

3.2 WORKFLOW

Our workflow consists of five steps, as illustrated in **Figure 3.1**. These steps need to be performed in sequence. Each step relies on the result from the previous steps. In this section, we will explain how each step works and provide some examples to illustrate the process.

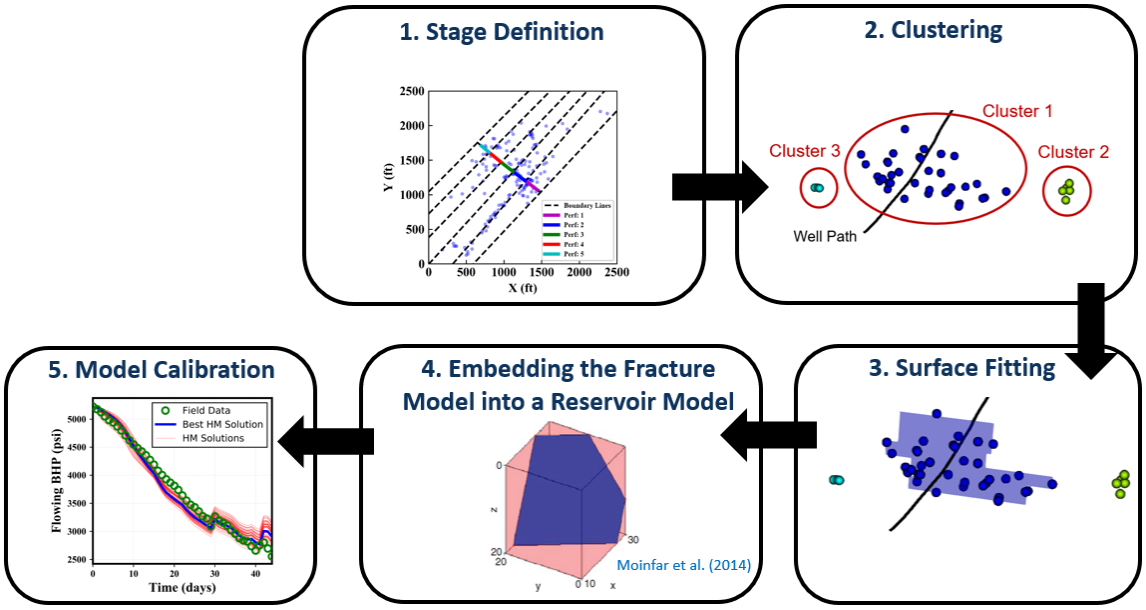

Figure 3.1: General workflow to generate a calibrated fracture model using microseismic events.

3.2.1 Step 1: Stage definition

As explained in Chapter 2, microseismic events from different stages often overlap. Therefore, our first task is to resolve this overlapping issue. We rely on the perforation intervals to help us identify the boundary of each stage. Then, we re-assign each microseismic event to appropriate stage. Below is the workflow we use.

1. We plot the microseismic events in the x-y plane, **Figure 3.2** (a).
2. We discard the stage assignment provided from the field data. In Figure 3.2 (b), all microseismic events changed to the same color.
3. We identify where each perforation interval starts and ends in the wellbore trajectory. Figure 3.2 (b) shows perforation intervals with different colors.
4. We draw perpendicular lines considering the perforation intervals and the slope of the well path at each boundary point, Figure 3.2 (b). Because we are using actual

wellbore trajectories, the slope of the well path will be different at each point in the well path. We use central differencing scheme to approximate the slope at each point in the well path. The slope of the boundary lines can be identified using the following simple relationship.

$$Slope\ of\ a\ boundary\ line = \frac{-1}{Slope\ of\ the\ well\ path}$$

5. We assign each microseismic event to the appropriate stage based on its location with respect to the boundary lines, Figure 3.2 (c). We know the equation of each boundary line from the previous step. Then, we evaluate each microseismic event by calculating the y-coordinate predicted from each line equation. After that, we compare the actual y-coordinate of the microseismic event with the predicted from all the boundary lines. Identifying the closest two predictions can tell us where is the microseismic event located with respect to the boundary lines.

The above workflow guarantees that there is no overlapping between the stages and prepare the microseismic data for the next step.

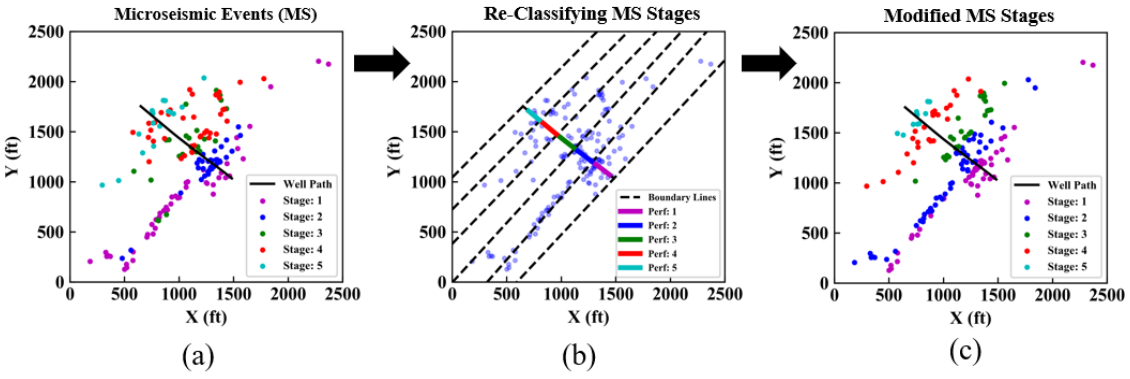

Figure 3.2: Assigning microseismic events to the appropriate stage based on the location of the perforation intervals. (a) Original microseismic events. (b). Drawing perpendicular boundary lines between the stages. (c) Microseismic events are re-assigned to the appropriate stage.

3.2.2 Step 2: Clustering

As explained in Chapter 2, microseismic events are weak signals and contain noise from different sources. Therefore, reducing the noise level in the data before building the fracture model is important. We use a clustering method known as density-based spatial clustering of application with noise (DBSCAN). DBSCAN uses the notion of density to identify clusters within a dataset (Ester et al. 1996). DBSCAN tries to model human's intuition to identify groups based on density. For example, we can easily recognize the clusters and noise in **Figure 3.3** without performing any analysis because the density inside the clusters is much higher than outside.

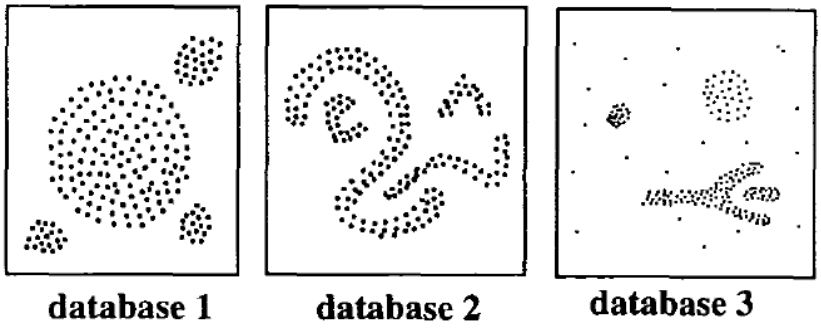

Figure 3.3: Simple examples that illustrates the basis of DBSCAN (Ester et al. 1996).

DBSCAN uses two input parameters from the user: epsilon (Eps) and minimum number of points (MinPts) and follows the workflow below.

1. DBSCAN examines each point in the dataset by drawing a circle around it with a radius Eps.
2. DBSCAN counts the number of points inside the circle.
3. DBSCAN classifies the data point into three categories based on the number of points from the previous step, **Figure 3.4**.

- If the number of points is greater than MinPts, then the point is classified as a core point.
- If the number of points is less than MinPts, then the point is classified as noise.
- If the number of points is less than MinPts, however there is at least one core point inside the circle, then the point is classified as a border point.

4. The core points and border points are grouped together to form a cluster. Noise points are not placed in any cluster.

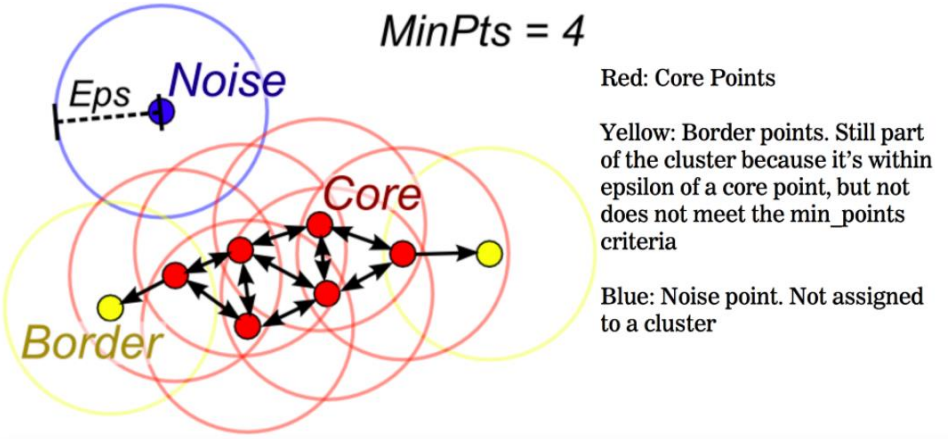

Figure 3.4: DBSCAN workflow. DBSCAN classifies the points in the dataset as core, border, or noise based on the number of points found in the investigation circle (Lutins 2017).

The original DBSCAN workflow was designed for 2D data. However, it works in a similar way for 3D data. **Figure 3.5** illustrates the application of DBSCAN with microseismic events in one stage. If DBSCAN finds more than one cluster in a stage, we choose only the cluster that is closest to the wellbore.

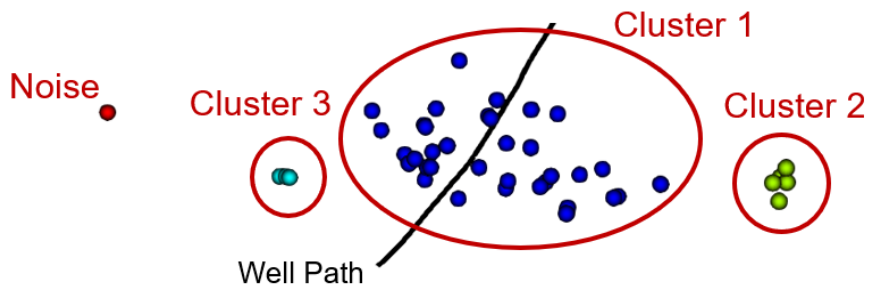

Figure 3.5: Clusters identified by DBSCAN in one stage. Input parameters are Eps = 250 ft and MinPts = 3.

3.2.3 Step 3: Surface fitting

Now we fit a fracture plane into the chosen cluster from the previous step. Our guess function is the equation of a plane, **Equation 1**. The unknowns are the coefficients a, b, and c in Equation 1. We need to find the coefficients that would minimize our error function, **Equation 2**. We defined the error as the summation of all the squared distances in the y direction between the actual microseismic events and the fracture plane. We use the squared distance instead of distance to account for the error in both sides of the plane.

To solve for the coefficients, we take the partial derivative of the error function with respect to a, b, and c. Then, we set the partial derivatives to zero and rearrange them to get a system of three linear equations, **Equation 3**.

$$g(x,y) = y = a + bx + cz \tag{1}$$

$$E(g) = \sum_{k=1}^{m} [g(x_k, z_k) - y_k]^2 \tag{2}$$

$$\begin{bmatrix} m & \sum x_k & \sum z_k \\ \sum x_k & \sum {x_k}^2 & \sum x_k z_k \\ \sum z_k & \sum x_k z_k & \sum {z_k}^2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sum y_k \\ \sum x_k y_k \\ \sum y_k z_k \end{bmatrix} \tag{3}$$

By solving Equation 3, we now have the equation of the plane and can plot it over the microseismic events, **Figure 3.6**. Also, we can obtain the dip angle (θ), azimuth (α), and the coefficient of determination (R^2) using **Equations 4-6**, respectively. Derivation of Equations 4 and 5 are available in **Appendix A**. The coefficient of determination is used to evaluate how well a model matches with measured data.

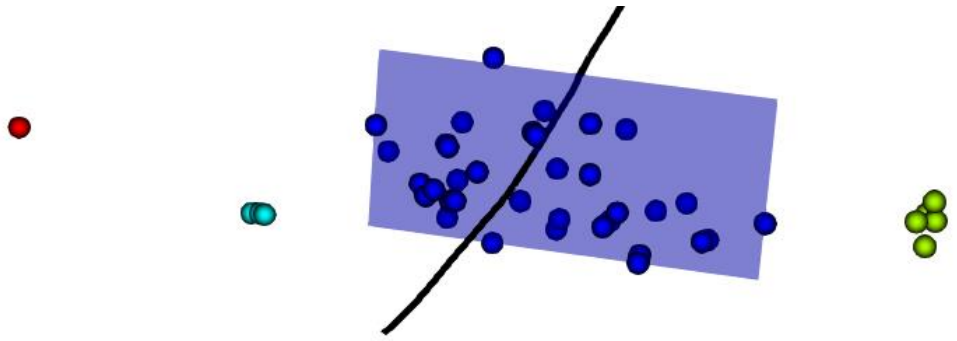

Figure 3.6: Fitting a fracture plane over the microseismic events after solving for the coefficients a, b, and c.

$$\theta = cos^{-1}\left[\frac{c}{\sqrt{b^2+c^2+1}}\right] \qquad for\ c > 0$$
$$\theta = cos^{-1}\left[\frac{-c}{\sqrt{b^2+c^2+1}}\right] \qquad for\ c < 0 \tag{4}$$

$$\alpha = tan^{-1}\left[\frac{-1}{b}\right] \qquad for\ b > 0$$
$$\alpha = 180 - tan^{-1}\left[\frac{-1}{b}\right] \qquad for\ b < 0 \tag{5}$$

$$R^2 \overset{\text{def}}{=} 1 - \frac{\sum_{k=1}^{m}(z_k - g(x_k, y_k)^2}{\sum_{k=1}^{m}(z_k - \bar{z})^2} \tag{6}$$

3.2.3.1 Finding the well-fracture intersection

One important information we need to know is the intersection point between the fracture and the wellbore. We will use this information later in the model calibration step. In this section, we explain how to calculate the well-fracture intersection point.

First, we find the two points in the well path that bound the fracture plane. Usually the well path is provided as discrete points in the three-dimensional space, **Figure 3.7** (a). Considering the schematic in Figure 3.7 (b), we use the following workflow.

1. We calculate Δy for each point in the well path.

$$\Delta y = y_{well} - y_{plane}$$

2. We find the closest well-path point to the plane, which is the point with the smallest Δy in absolute value.
3. We find the second point on the other side of the plane. The second point will be next to the closest point and have opposite Δy sign.

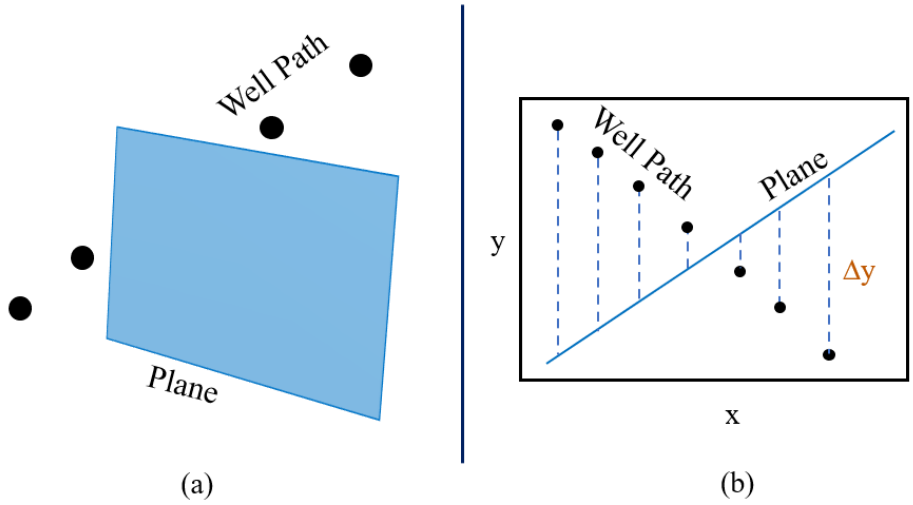

Figure 3.7: Finding the two points in the well path that bound the fracture plane. (a) Well path and fracture plane in 3D. (b) Well path and fracture plane in 2D.

After identifying the two closest points in the well path, we can generate the equation of a line in 3D that connects the two points using **Equations 7-9**. The intersection point should satisfy both the equation of the line and the equation of the plane, **Equation 10** and **Figure 3.8**. Therefore, we have a system of four linear equations with four unknowns (t, x, y, and z). The system can be solved for the intersection point using **Equation 11**.

$$x = x_0 + t\,\Delta x \tag{7}$$

$$y = y_0 + t\,\Delta y \tag{8}$$

$$z = z_0 + t\,\Delta z \tag{9}$$

$$y = a + bx + cz \tag{10}$$

$$\begin{bmatrix} -\Delta x & 1 & 0 & 0 \\ -\Delta y & 0 & 1 & 0 \\ -\Delta z & 0 & 0 & 1 \\ 0 & -b & 1 & -c \end{bmatrix} \begin{bmatrix} t \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ a \end{bmatrix} \tag{11}$$

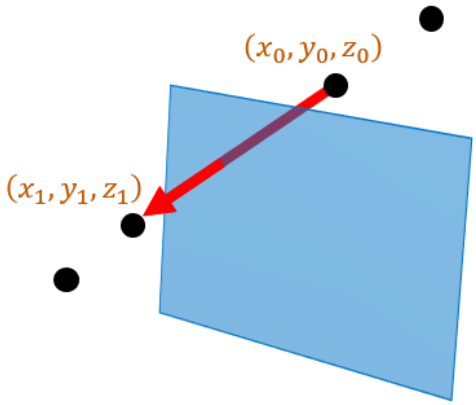

Figure 3.8: Fining the intersection point between the well path and the fracture plane.

3.2.3.2 Defining fracture boundaries

The equation of a plane does not provide any information about the plane boundary. It represents an infinite plane in the 3D space. We can restrict the fracture plane to a rectangular shape over the microseismic events as in Figure 3.6. However, actual fractures might have irregular shapes, especially in heterogenous reservoirs such as shale reservoirs. Therefore, we try to interpret the fracture shape from the distribution of the microseismic events as in **Figure 3.9**. We combine several rectangular planes to build one polygonal plane using the following workflow.

1. We specify the number of microseismic events we want in each rectangular plane and the maximum number of planes to combine in one polygon.
2. Using the input parameters from step 1 and the total number of microseismic events, we calculate how many rectangular planes we should use (n).
3. We sort the microseismic events with respect to their depth and divide them into (n) number of sets, **Figure 3.10** (b). To make sure that the rectangular planes are connected, we duplicate one microseismic event in two neighboring sets. For example, the last microseismic event in set 1 will be the first microseismic event in set 2.
4. We find the maximum and minimum x and z coordinates in each set and calculate the y coordinate using the equation of the plane.
5. We combine (n) number of planes in one polygon, Figure 3.10 (c).

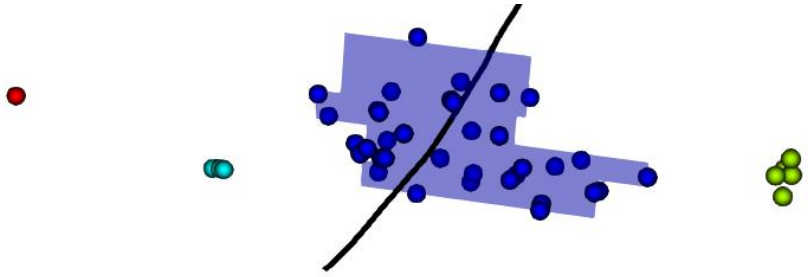

Figure 3.9: Fracture plane fit with polygonal shape.

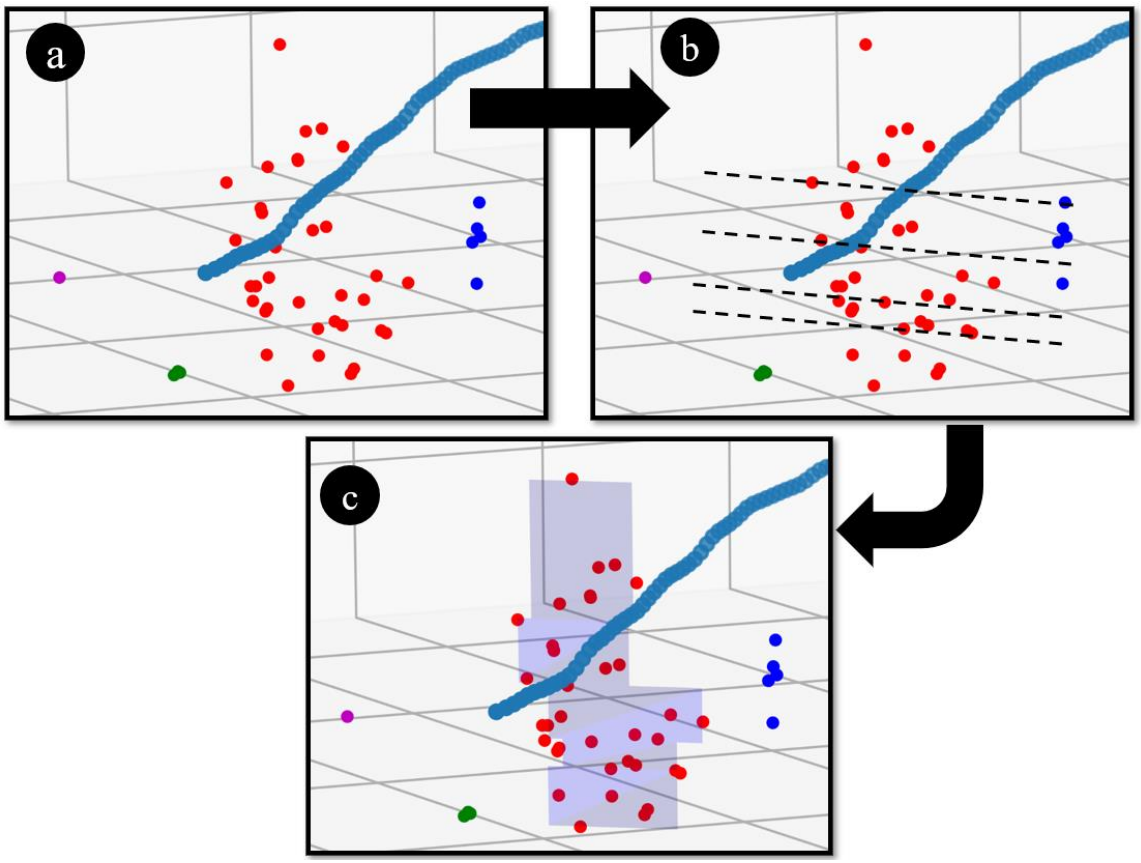

Figure 3.10: Workflow that combines several rectangular fracture planes into one polygonal fracture plane. (a) Microseismic events in one stage. (b) Dividing the microseismic events into “n” number of sets. (c) Combining “n” number of rectangular planes into one polygonal plane.

3.2.3.3 Fitting multiple fractures per each stage

Here we introduce a method to fit more than one fracture per stage. We use DBSCAN again to identify clusters after fitting the fracture plane. The workflow below explains the method.

1. We calculate the distance in the y direction (Δy) between the microseismic events and the fracture plane, **Figure 3.11**. We can consider this distance as the error because it is the difference between the actual data and the model prediction.
2. We plot the error distribution as a histogram, **Figure 3.12** (1), or one-dimensional scatter plot, Figure 3.12 (2).
3. We identify clusters by applying DBSCAN to the error distribution, **Figure 3.13**. As explained before, we need to specify the two DBSCAN parameters (Eps and MinPts).
4. We discard clusters with too few microseismic events (e.g. less than 4).
5. We fit a new fracture for each remaining cluster, **Figure 3.14**.

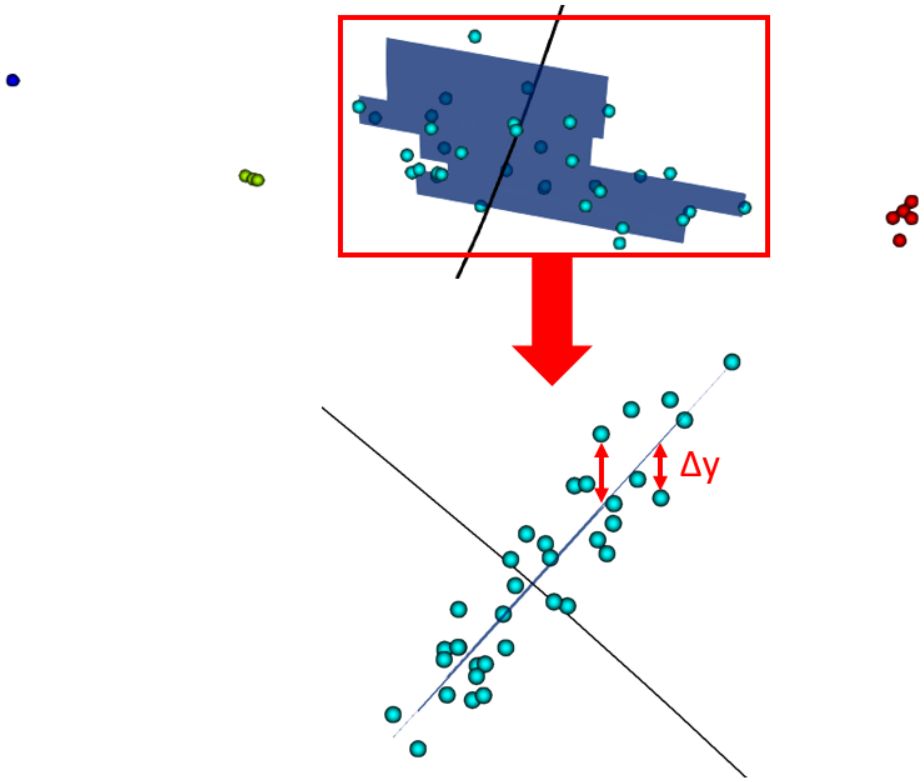

Figure 3.11: Calculating the error distribution for each microseismic event.

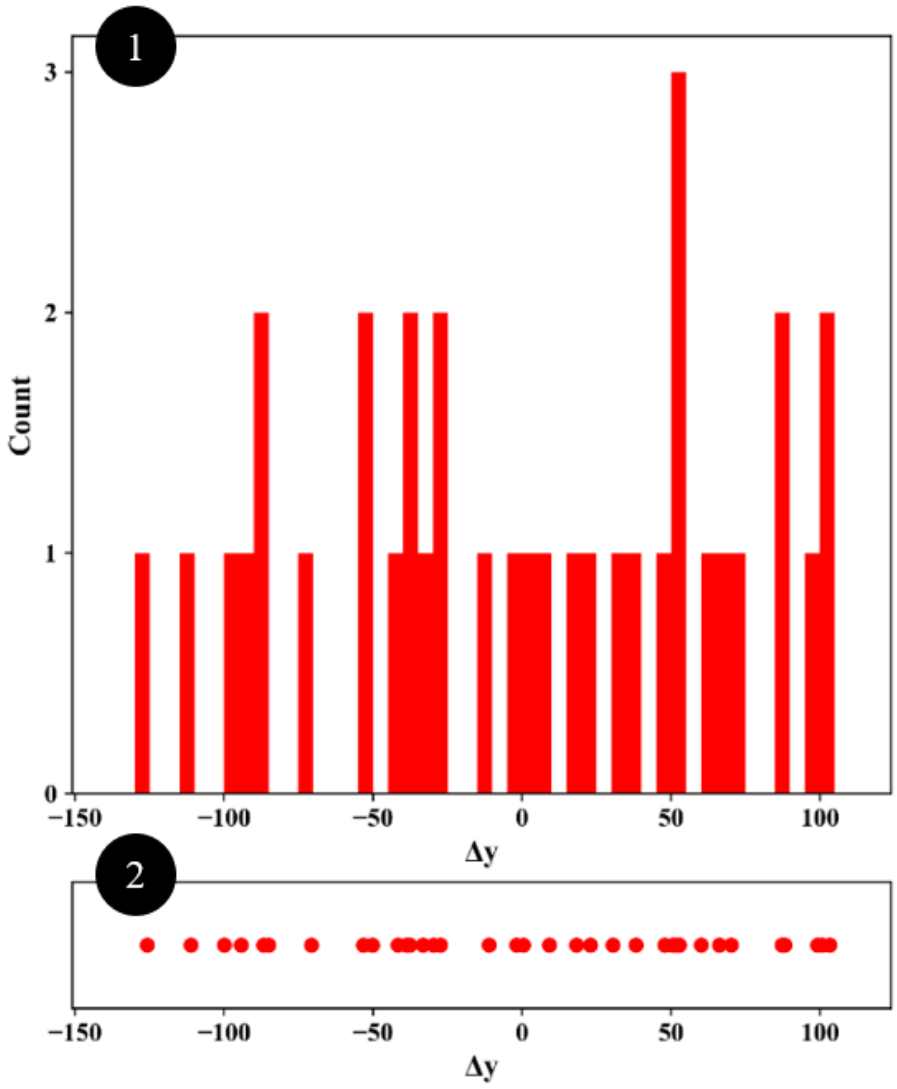

Figure 3.12: Error distribution after fitting the first fracture, which is plotted as a histogram in (1) and as a one-dimensional scatter plot in (2).

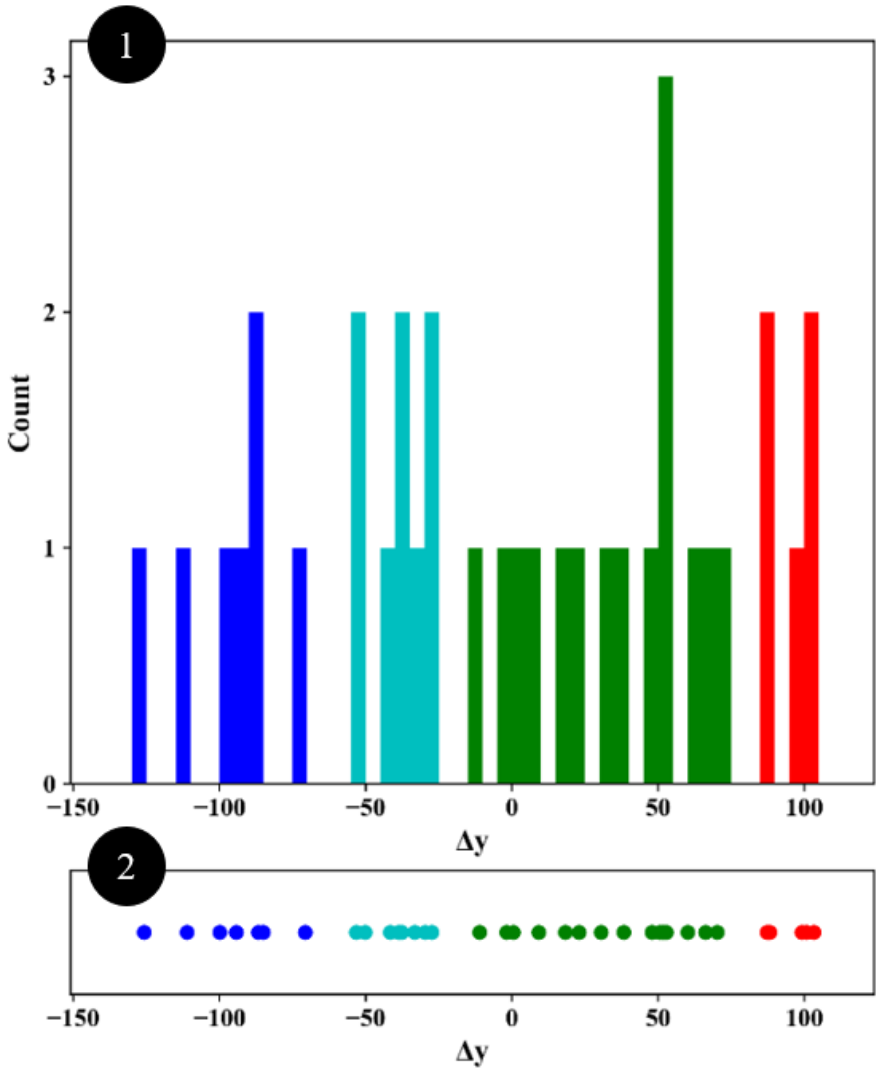

Figure 3.13: Applying DBSCAN to the error distribution. The error distribution is plotted as a histogram in (1) and as one-dimensional scatter plot in (2). DBSCAN parameters Eps = 15 ft and MinPts = 2.

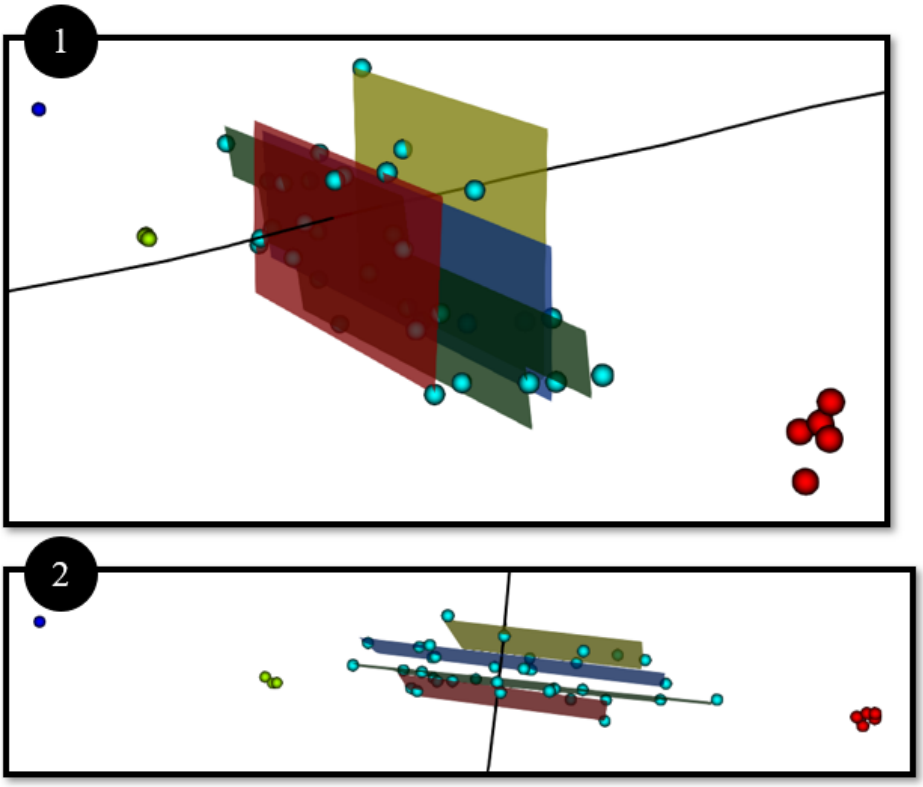

Figure 3.14: Fitting more than one fracture in a single stage. (1) Side view. (2) Top view.

3.2.4 Step 4: Embedding the fracture model into a reservoir model

In this step, we incorporate the fracture model that we built into a reservoir simulation model. We have discussed in Chapter 2 several methods used to include the effect of fractures in reservoir simulation models. The embedded discrete fracture model (EDFM) method is useful in our application. Our fracture model will contain fractures with different dip angles, shapes, and sizes. EDFM is proficient in modeling such fractures because it discretizes the fractures and move them to a different domain.

In addition, we will run the reservoir simulation model many times in the model calibration step. Therefore, using an efficient method such as EDFM is important. Also,

EDFM can be coupled with most conventional reservoir simulator, which means that we can implement our workflow in many reservoir simulation platforms.

3.2.5 Step 5: Model calibration

At this point in the workflow, we have already built a fracture model. However, the two input parameters in DBSCAN introduces a new source of uncertainty. Using different Eps and MinPts may result in a different fracture model. To manage the uncertainty, we added this final step to calibrate our model and perform history matching with fluid production and bottom hole pressure from field data.

3.2.5.1 Scaling factors

To calibrate the fracture model, we introduce two scaling factors. The first factor reduces the fracture half-length and the second reduces the fracture height. In all cases, we keep the dip and azimuth angles the same as the original fracture model. Then, we add the two scaling factors to the uncertain parameters in an assisted history matching (AHM) workflow. AHM helps us identify which scaling factor is most likely to match the field production data.

One challenge we encounter in cutting the fractures with the scaling factors is the risk of losing the connection to the wellbore. The fractures in our model have arbitrary polygonal shapes and the well-fracture intersection points may not be in the center of the fractures. **Figure 3.15** shows an example of how we lose well-fracture connection point when using 0.25 for both scaling factors.

However, we expect hydraulic fractures to be connected to the wellbore. Also, if the wellbore loses connection to many of the fractures, the simulated production rates will

be significantly affected. To resolve this issue, we developed the following workflow to cut the fractures while keeping them connected to the wellbore.

1. We identify the well-fracture intersection point, **Figure 3.16** (a). We have already discussed how we obtain the intersection point in step 2.
2. We reduce the fracture height (or half-length) from one side only so that the intersection point becomes in the middle of the fracture, Figure 3.16 (b) and (d). We perform this step by calculating the shortest vertical (or horizontal) distance from the intersection point to the end of the polygon, **Figure 3.17** and **Figure 3.18**. Then, the calculated distance will be used on other side to decide where to cut the fracture.
3. We reduce the fracture height (or half-length) from both sides of the intersection point equally until the targeted (or half-length) is achieved, Figure 3.16 (c) and (e). Our cutting function relies on vector analysis in the three-dimensional space, see **Appendix B**.

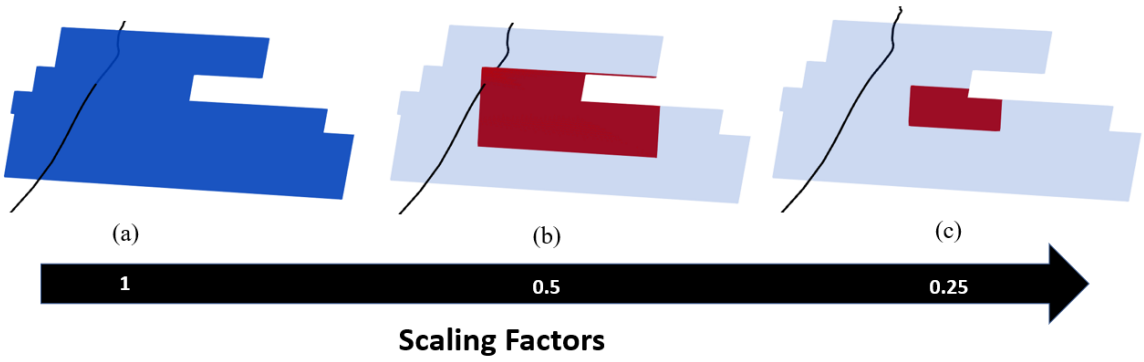

Figure 3.15: Example of losing the well-fracture connection point. (a) Original fracture before applying any scaling factors. (b) New fracture (in red) after using 0.5 for both scaling factors. (c) New fracture (in red) after using 0.25 for both scaling factors.

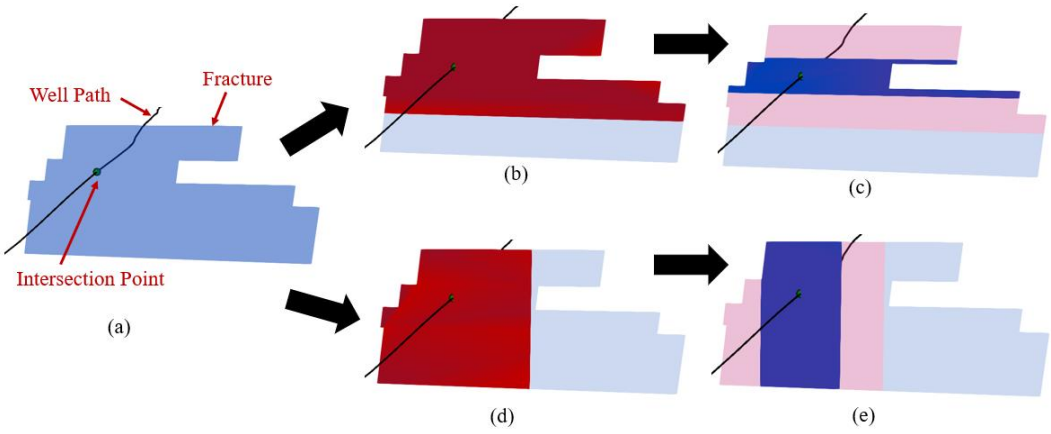

Figure 3.16: Reducing the fracture height and half-length using scaling factors. (a) Original fracture before applying any scaling factor. (b) & (d) Cutting the fracture from one side only so that the intersection point becomes in the middle of the fracture in the fracture height and half-length direction, respectively. (c) & (e) Cutting the fracture from both sides until the target fracture height and half-length, respectively, is achieved.

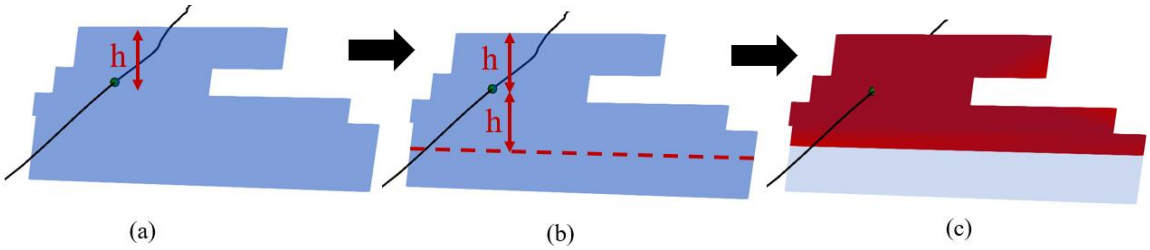

Figure 3.17: Cutting the fracture vertically from one side only to avoid losing the intersection point.

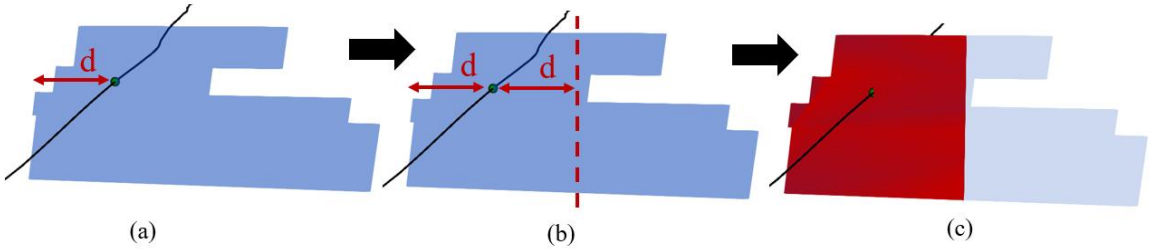

Figure 3.18: Cutting the fracture horizontally from one side only to avoid losing the intersection point.

3.2.5.2 Assisted history matching (AHM)

The two scaling factors introduced in the previous section are treated as uncertain parameters and included in an assisted history matching (AHM) workflow. We use an AHM workflow developed by Sun (2019), **Figure 3.19**. The goal of the AHM workflow is to find many history matching solutions that are different from one another. Therefore, the workflow tries to balance between exploring the uncertainty domain and pursuing history matching solutions. We describe the AHM workflow in the steps below.

1. Using the uncertain parameters' ranges, we generate 25 initial samples. We select mainly the maximum and minimum values for each uncertain parameter and some average values as well. We can also generate the samples using two level factorial design or Latin hypercube sampling.
2. We run the reservoir simulation model for these 25 initial samples and calculate the mean square error (MSE) for the production rates and bottom-hole pressure for each sample. We also calculate the weighted summation (MSE Sum) of the all the MSE's for each sample based on pre-defined weighting factors.
3. We generate 10,000 random samples using the ranges defined for each uncertain parameter.

4. We estimate the results of each sample without running the simulation model. We use instead the K-Nearest Neighboring (KNN) method as our proxy model. We use the result obtained from the initial design to estimate the results for the new sample. We calculate the Euclidean distance between the new sample to all the stored sample from the initial design and pick the three closest samples to estimate the result for the new sample.
5. We sort the random samples based on their estimated MSE Sum and pick the top 3000 samples for the next step.
6. We use a diverging sample function to select 200 samples. The objective of this function is to select samples that are far away from each other and far away from the samples in the initial design.
7. We sort again the selected 200 samples based on their estimated MSE Sum and pick the top 10 samples for reservoir simulation.
8. We repeat steps 3-7 for each iteration. We usually use 30 to 50 iterations.

The number of the selected samples in steps 5 and 6 can be adjusted for each iteration to improve the sampling strategy, **Figure 3.20**. To reduce exploring the uncertainty domain and increase the number of history matching solutions, we should reduce the number of selected samples in step 5 and increase the number of selected samples in step 6.

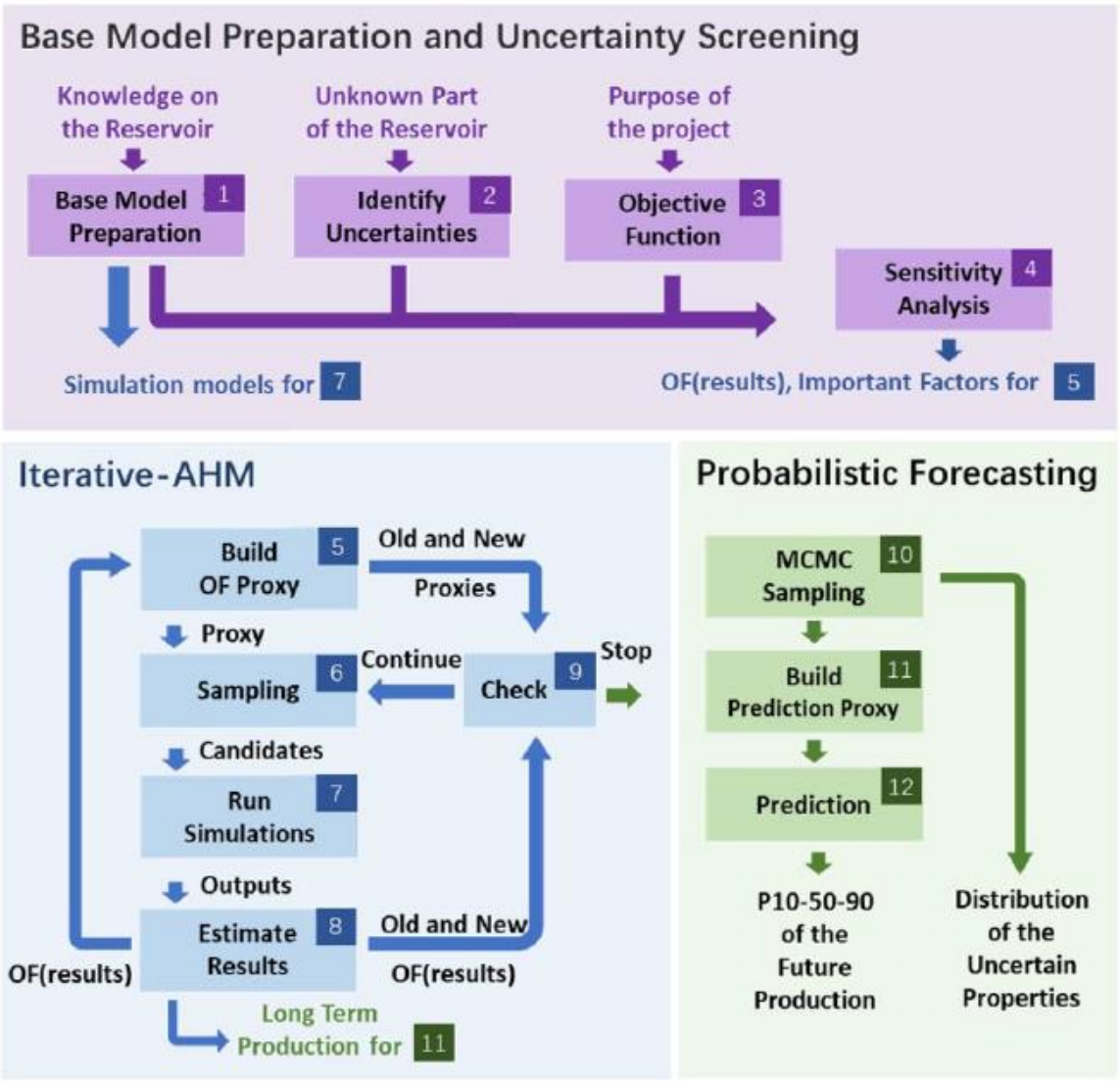

Figure 3.19: Assisted history matching workflow (Sun 2019).

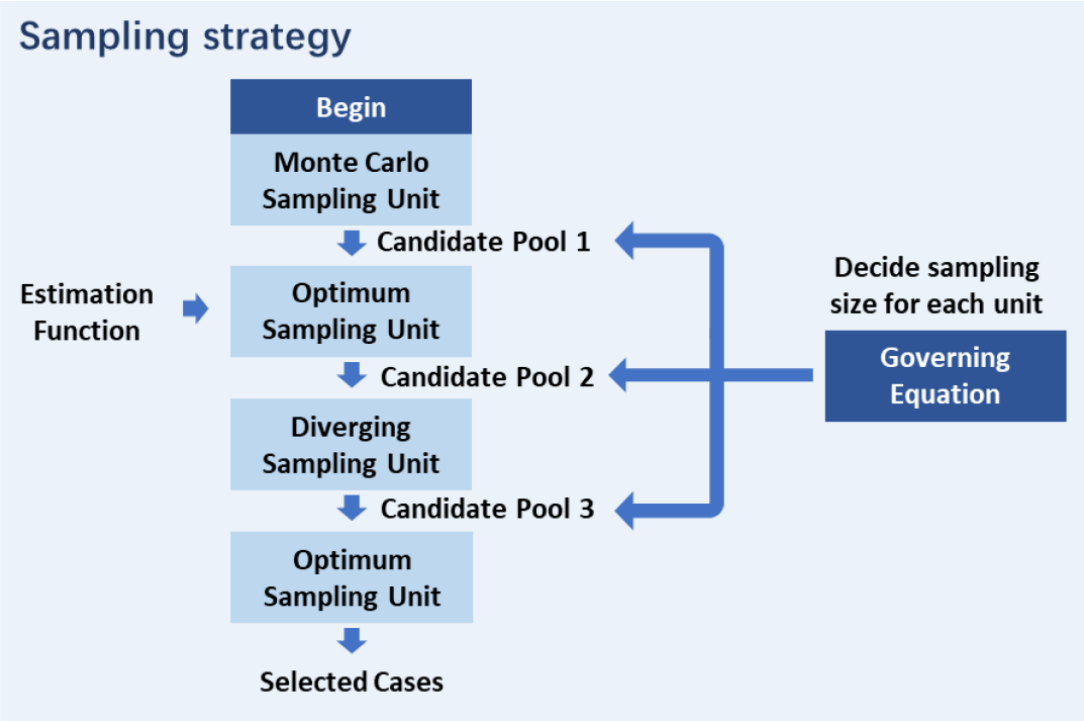

Figure 3.20: Changing the number of selected samples in each sampling unit to improve the sampling strategy (Sun 2019).

3.2.5.3 Converting a Deviated Well into a Horizontal Well

Most wellbores are deviated and not completely horizontal. However, including a deviated wellbore in a simple reservoir simulation model creates more complications. We need to increase the reservoir thickness to accommodate the deviated wellbore. Increasing the reservoir thickness is not recommended because it reduces the accuracy of the reservoir model. A simple solution is to convert the deviated wellbore into a horizontal wellbore in the simulation model. In the following workflow, we describe how we transfer the fractures and intersection points from the actual deviated wellbore to the horizontal wellbore in the simulation mode.

1. For each fracture, we identity a reference point along the well path, **Figure 3.21**. We use the intersection points we calculated in step 3 as our reference points.

2. We calculate the change in x, y, and z directions (Δx, Δy, and Δz) between the reference point and each vertex in the polygon.
3. We calculate and record the distances between the reference points.
4. We transfer the reference points to the new horizontal wellbore while preserving the distances between the reference points.
5. We recreate the fracture polygons using the values of Δx, Δy, and Δz from step 2.

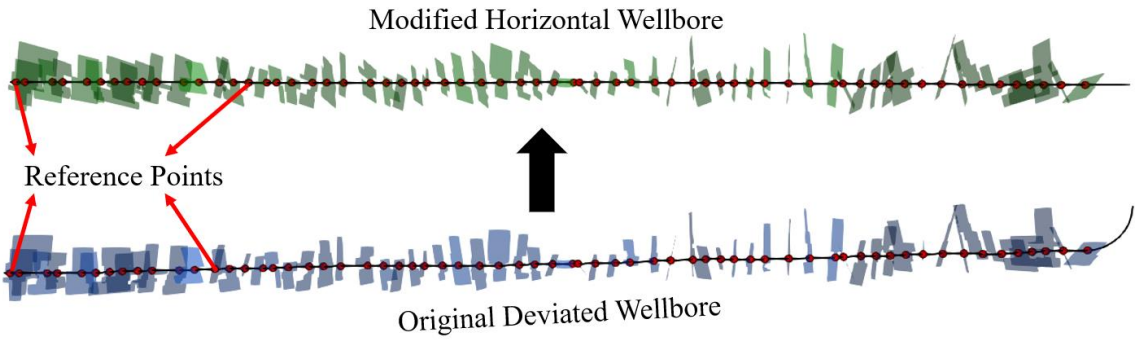

Figure 3.21: Converting a deviated wellbore into a horizontal wellbore using reference points.

Chapter 4: Converting Wellhead Pressure to Bottom Hole Pressure in Gas Wells

In this chapter, we explore a different topic compared to the rest of thesis. To calibrate reservoir simulation models, we try to match production and bottom hole pressure data from the field. Production data are usually available and easy to obtain because production rates are typically measured at surface. However, bottom hole pressure measurements are harder to obtain and more expensive. Measuring bottom hole pressure involves lowering a gauge downhole for a short period of time or installing permanent gauges in the wellbore. Both options are expensive and require a lot of resources.

Therefore, companies prefer to measure pressure at the wellhead and try to estimate the bottom hole pressure. For wellbores filled with a single liquid phase (water only or oil only), converting wellhead pressure to bottom hole pressure is simple. We only need to know the liquid density, which helps us calculate the weight of the liquid column at a certain depth. However, for wellbores filled with more than one phase, the problem becomes more complicated for two reasons. First, there is no direct method to predict the location of the interfaces using surface data only. Second, if one of the phases in the wellbore is gas, then calculating the weight of the gas column is more complicated. Gas is compressible and its density changes with pressure and temperature.

In this chapter, we introduce a novel method that converts static wellhead pressure to bottom hole pressure in wellbores filled with gas and water. The gas column is discretized, and pressure is calculated in each segment using Newton-Raphson method. In addition, the gas-water interface is treated as an unknown in the Newton-Raphson system.

4.1 Assumptions

The followings are the main assumptions we used to develop the method.

1. Considering a well producing gas and water at a gas water ratio GWR, we assume that both gas and water will remain in the wellbore after shutting the well from surface. In other words, we assume that gas and water will not flow back to the formation because of pressure difference between the wellbore and the formation.
2. We assume that the temperature in the wellbore is constant and equal to the average temperature between surface and the reservoir temperature.
3. After discretizing the gas column, we assume that the pressure at each segment is constant.
4. The wellhead pressure was stable at the time of measurement.
5. The gas and water in the wellbore are at equilibrium and the gas-water interface is stable at one depth.
6. The wellbore has a constant radius.

4.2 METHOD

4.2.1 Mathematical Model

Considering the schematic in **Figure 4.1**, the gas and water volumes are evaluated at two different conditions: surface and wellbore conditions. The gas columns in the wellbore is divided equally into (n) segments. The unknowns are the pressure at each segment (p_k) and the depth of the gas-water interface (D_1). Therefore, we need to solve for (n+1) unknowns, which requires (n+1) equations. The equivalent volume of water and each segment in the gas column is evaluated at surface.

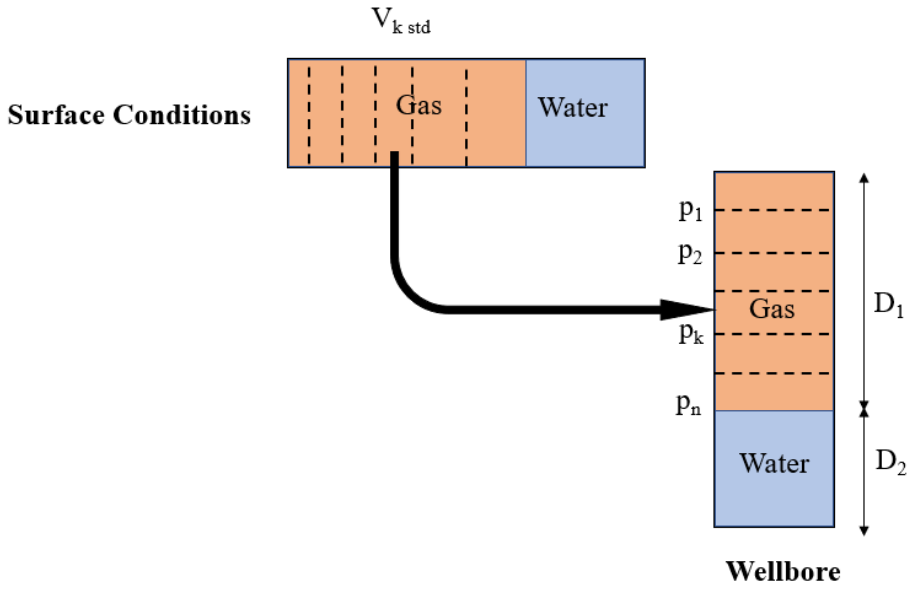

Figure 4.1: Schematic to convert wellhead pressure to bottom hole pressure in wellbores filled with gas and water.

The gas water ratio (GWR) at surface conditions is known from field production data. GWR is equal to the ratio of gas volume at standard conditions (V_{std}) to the water volume at standard conditions (V_w), **Equation 12**. The gas segments have the same volumes in the wellbore because the gas column was discretized uniformly. However, after converting these gas segments to surface conditions, they will have different volumes because each segment has different pressure in the wellbore. The deepest gas segment in the wellbore will occupy a larger volume at surface compared to the shallowest gas segment.

The gas volume at standard condition is equivalent to the summation of all gas segment volumes at surface conditions, **Equation 13**. Water volume at surface conditions is the same as the water volume downhole, assuming a formation volume factor of 1.

Therefore, the water volume can be written in terms of the height of the water column (D_2), **Equation 14**.

$$GWR = \frac{V_{std}}{V_w} \tag{12}$$

$$GWR = \frac{1}{V_w}[V_{1\,std} + V_{2\,std} + \cdots + V_{n\,std}] \tag{13}$$

$$GWR = \frac{1}{\pi r^2 D_2}[V_{1\,std} + V_{2\,std} + \cdots + V_{n\,std}] \tag{14}$$

Next, using the ideal gas law with z-factor correction, we relate the volume of a gas segment at surface to the volume of a gas segment at the wellbore, **Equations 15 and 16**. The volume of a gas segment at the wellbore (V_k) can be written in terms of the height of the gas column (D_1), **Equation 17**. Only three parameters (D_1, p_k, and $V_{k\,std}$) are unknown in Equation 17.

$$\frac{p_1 V_1}{z_1 T_1} = \frac{p_2 V_2}{z_2 T_2} \tag{15}$$

$$\frac{p_{std} V_{k\,std}}{z_{std} T_{std}} = \frac{p_k V_k}{z_k T} \tag{16}$$

$$V_{k\,std} = \frac{z_{std} T_{std}}{p_{std} T}\left(\frac{p_k}{z_k}\right)\left(\pi r^2 \frac{D_1}{n}\right) \tag{17}$$

Combining Equations 14 and 17, we can derive an expression that has (n+1) unknowns, **Equation 18**. The unknowns are the pressure in each gas segment in the wellbore (p_1, p_2, …, p_n) and the height of the gas column (D_1). We can generate the other (n) equations using **Equation 19**. The derivation of Equation 19 is available in Cullender and Smith (1956) and in **Appendix C**.

The objective of Equation 19 is to convert pressure in a gas column from one depth to another. In our application, we apply Equation 19 for each segment in the gas column.

First, we apply the equation from wellhead pressure (WHP) to the pressure at the end of the first segment (p_1), **Equation 20**. Then, we apply the equation for the remaining segments using **Equations 21 and 22**.

$$f_1 = GWR + \frac{z_{std} T_{std} D_1}{n\, p_{std}\, T\, (D_1 - D)} \left[\frac{p_1}{z_1} + \frac{p_2}{z_2} + \cdots + \frac{p_n}{z_n}\right] = 0 \tag{18}$$

$$\frac{\gamma_g\, D}{53.351} = \int_{p_w}^{p_s} \frac{T\, z}{p}\, dp \tag{19}$$

$$f_2 = \frac{-\gamma_g\, D_1}{53.351\, n} + \int_{WHP}^{p_1} \frac{T\, z}{p}\, dp \tag{20}$$

$$f_k = \frac{-\gamma_g\, D_1}{53.351\, n} + \int_{p_{k-2}}^{p_{k-1}} \frac{T\, z}{p}\, dp \tag{21}$$

$$f_{n+1} = \frac{-\gamma_g\, D_1}{53.351\, n} + \int_{p_{n-1}}^{p_n} \frac{T\, z}{p}\, dp \tag{22}$$

4.2.2 Numerical Solution

We use Newton-Raphson method to solve the (n+1) equations. We construct the Jacobian matrix by taking the partial derivatives of each equation with respect to each unknown, **Equations 23 to 28**. The following explains the process we follow.

1. We start with an initial guess for the unknowns (D_1, p_1, p_2, …, p_n).
2. Using the initial guess, we calculate the functions ($f_1, f_2, \ldots, f_{n+1}$) and the Jacobina matrix (J).
3. We solve the equation $J\, \delta X = -f$ for the vector δX. The vector f is a column matrix that consists of f_1 to f_{n+1}.
4. We add the vector δX to our initial guess and use the result as our next guess.
5. We repeat the steps above until the solution converges.

$$J = \begin{bmatrix} \frac{df_1}{dD_1} & \frac{df_1}{dp_1} & \frac{df_1}{dp_2} & \cdots & \cdots & \frac{df_1}{dp_n} \\ \frac{df_2}{dD_1} & \frac{df_2}{dp_1} & 0 & \cdots & \cdots & 0 \\ \frac{df_3}{dD_1} & \frac{df_3}{dp_1} & \frac{df_3}{dp_2} & 0 & \cdots & 0 \\ \vdots & & & & & \\ \vdots & & & & & \\ \frac{df_n}{dD_1} & 0 & 0 & 0 & \frac{df_n}{dp_{n-1}} & \frac{df_{n+1}}{dp_n} \end{bmatrix} \tag{23}$$

$$\frac{\partial f_1}{\partial D_1} = \frac{-z_{std} T_{std}\, D}{n\, p_{std}\, T\, (D_1 - D)^2} \left[\frac{p_1}{z_1} + \frac{p_2}{z_2} + \cdots + \frac{p_n}{z_n}\right] \tag{24}$$

$$\frac{\partial f_1}{\partial p_k} = \frac{z_{std} T_{std} D_1}{n\, P_{std}\, T\, (D_1 - D)} \left[\frac{1}{z_k} - \frac{p_k}{{z_k}^2} \frac{dz_1}{dp_1}\right] \tag{25}$$

$$\frac{\partial f_k}{\partial D_1} = \frac{-\gamma_g}{53.351\, n}, \quad k = 2,\, 3,\, \ldots n+1 \tag{26}$$

$$\frac{\partial f_k}{\partial p_{k-1}} = \frac{T\, z_{k-1}}{p_{k-1}}, \quad k = 2,\, 3,\, \ldots n+1 \tag{27}$$

$$\frac{\partial f_k}{\partial p_{k-2}} = \frac{T\, z_{k-2}}{p_{k-2}}, \quad k = 3,\, 4, \ldots,\, n+1 \tag{28}$$

4.2.2.1 Integral Estimation

We estimate the integral in Equations 20 to 22 using a six-step trapezoidal rule, **Equation 29**. The six-step trapezoidal rule was one of the methods recommended by Aziz (1967). Solving the integral analytically is challenging because the z-factor is a function of pressure and temperature. Therefore, estimating the integral numerically is more convenient. The trapezoidal rule discretizes the pressure range and uses trapezoids to estimate the area under the curve, **Figure 4.2**.

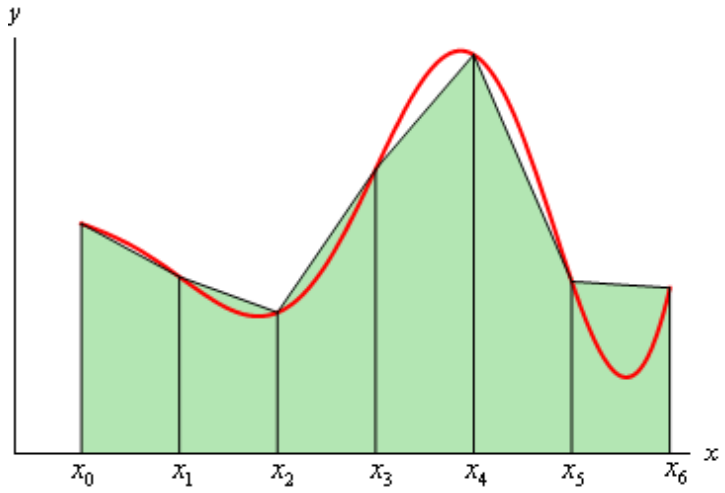

Figure 4.2: Estimating the area under the curve using a six-step trapezoidal rule (Dawkins 2020).

$$\int_{p_w}^{p_s} y\ dp \approx \frac{p_s - p_w}{6}\left[\frac{1}{2}y_0 + y_1 + y_2 + y_3 + y_4 + y_5 + \frac{1}{2}y_6\right] \tag{29}$$

$$y = \frac{T\,z}{p}$$

4.2.2.2 Z-Factor Calculations

Dranchuk and Abou-Kassem (1975) provide a correlation to calculate the compressibility factor (z-factor), **Equations 30 to 36**. To calculate the z-factor form Equation 30, we need the parameter (ρ_r). However, the z-factor is also needed to calculate the parameter (ρ_r), Equation 31. Therefore, we need to use an iterative method to solve for the z-factor.

$$z = 1 + \left(A_1 + \frac{A_2}{T_r} + \frac{A_3}{{T_r}^3} + \frac{A_4}{{T_r}^4} + \frac{A_5}{{T_r}^5}\right)\rho_r + \left(A_6 + \frac{A_7}{T_r} + \frac{A_8}{{T_r}^2}\right){\rho_r}^2 - A_9\left(\frac{A_7}{T_r} + \frac{A_8}{{T_r}^2}\right){\rho_r}^5 + A_{10}(1 + A_{11}{\rho_r}^2)\left(\frac{{\rho_r}^2}{{T_r}^3}\right)exp(-A_{11}{\rho_r}^2) \quad (30)$$

$$\rho_r = \frac{z_c p_r}{z T_r}, \qquad z_c = 0.27 \quad (31)$$

$$p_r = \frac{p}{p_{pc}} \quad (32)$$

$$T_r = \frac{T}{T_{pc}} \quad (33)$$

$$p_{pc} = 756.8 - 131.07\gamma_g - 3.6{\gamma_g}^2 \quad (34)$$

$$T_{pc} = 169.2 + 349.5\gamma_g - 74{\gamma_g}^2 \quad (35)$$

$$A_1 = 0.3265,\ A_2 = -1.0700,\ A_3 = -0.5339,\ A_4 = 0.01569,\ A_5 = -0.05165,\ A_6 = 0.5475,\ A_7 = -0.7361,\ A_8 = 0.1844,\ A_9 = 0.1056,\ A_{10} = 0.6134,\ A_{11} = 0.7210 \quad (36)$$

We use Newton-Raphson method to solve for the z-factor. All the terms in Equation 30 are moved to one side and can be written as a function of the z-factor $f(z)$, **Equation 37**. Newton-Raphson method tries to find the root of a function using the slope of the function at each iteration. Hence, we need to take the derivative of Equation 37 with respect to the z-factor, **Equation 38**. The following summarizes the steps we follow in the Newton-Raphson method.

1. We start with an initial guess of the z-factor $z^{(k)}$.
2. We calculate $f(z)$ and $f'(z)$ using Equations 37 and 38.
3. We find the next guess using **Equation 39**.
4. We repeat the above steps until the z-factor converges.

$$f(z) = z - 1 - \left(A_1 + \frac{A_2}{T_r} + \frac{A_3}{{T_r}^3} + \frac{A_4}{{T_r}^4} + \frac{A_5}{{T_r}^5}\right)\rho_r - \left(A_6 + \frac{A_7}{T_r} + \frac{A_8}{{T_r}^2}\right){\rho_r}^2 + A_9\left(\frac{A_7}{T_r} + \frac{A_8}{{T_r}^2}\right){\rho_r}^5 - A_{10}(1 + A_{11}{\rho_r}^2)\left(\frac{{\rho_r}^2}{{T_r}^3}\right)exp(-A_{11}{\rho_r}^2) \quad (37)$$

$$f'(z) = 1 + \left(A_1 + \frac{A_2}{T_r} + \frac{A_3}{{T_r}^3} + \frac{A_4}{{T_r}^4} + \frac{A_5}{{T_r}^5}\right)\frac{z_c p_r}{z^2 T_r} + 2\left(A_6 + \frac{A_7}{T_r} + \frac{A_8}{{T_r}^2}\right)\frac{{z_c}^2 {p_r}^2}{z^3 {T_r}^2} - 5A_9\left(\frac{A_7}{T_r} + \frac{A_8}{{T_r}^2}\right)\frac{{z_c}^5 {p_r}^5}{z^6 {T_r}^5} + 2A_{10}\left(\frac{{z_c}^2 {p_r}^2}{z^3 {T_r}^5} + \frac{A_{11} {z_c}^4 {p_r}^4}{z^5 {T_r}^7} - \frac{{A_{11}}^2 {z_c}^6 {p_r}^6}{z^3 {T_r}^9}\right)exp\left(-A_{11}\frac{{z_c}^2 {p_r}^2}{z^2 {T_r}^2}\right) \quad (38)$$

$$z^{(k+1)} = z^{(k)} - \frac{f\left(z^{(k)}\right)}{f'(z^{(k)})} \quad (39)$$

4.3 RESULTS

We applied the method to a gas well that has some water production. The water-gas ratio (WGR) history is presented in **Figure 4.3**. The decrease in water production in the early period is attributed to water flow back after a hydraulic fracturing job. Also, the input parameters for our workflow are included in **Table 4.1**.

Table 4.1: Input parameters to convert wellhead pressure to bottom hole pressure.

Parameter	**Value**	**Unit**
Wellbore total depth	10499	ft
Gas specific gravity	0.55	-
Reservoir temperature	214.7	F
Water pressure gradient	0.465	psi/ft
Number of discretization intervals in the gas column (n)	10	-

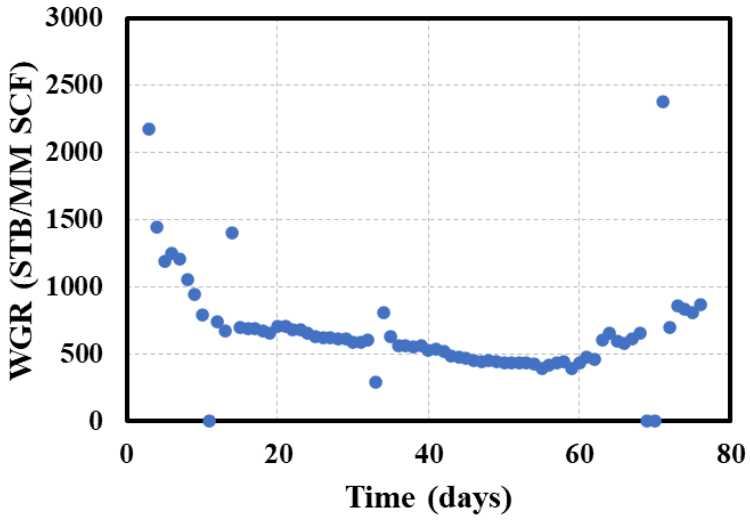

Figure 4.3: Water-gas ratio (WGR) history obtained from field production data.

The workflow explained in the previous section was repeated for each WGR value in the WGR history. **Figure 4.4** illustrates both the wellhead pressure (WHP) and the bottom hole pressure (BHP). The wellhead pressure was obtained from field data and the bottom hole pressure is the output of our workflow. We also compare our solution to bottom hole pressure values obtained by field correlation, **Figure 4.5**. Our solution was close the bottom hole pressure expected in the field.

Another output parameter from our workflow is the depth of the gas-water interface. Our initial guess is that the interface is in the middle of the wellbore. Then, after the solution converges, we record the final position of the gas-water interface. **Figure 4.6** shows how the gas-water interface depth change as the WHP and WGR change in the field data.

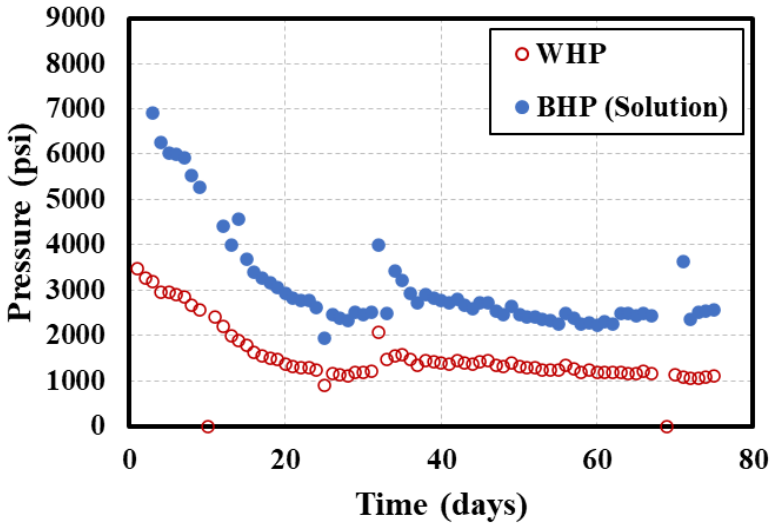

Figure 4.4: Wellhead pressure (WHP) history and the corresponding bottom hole pressure (BHP) obtained from the workflow.

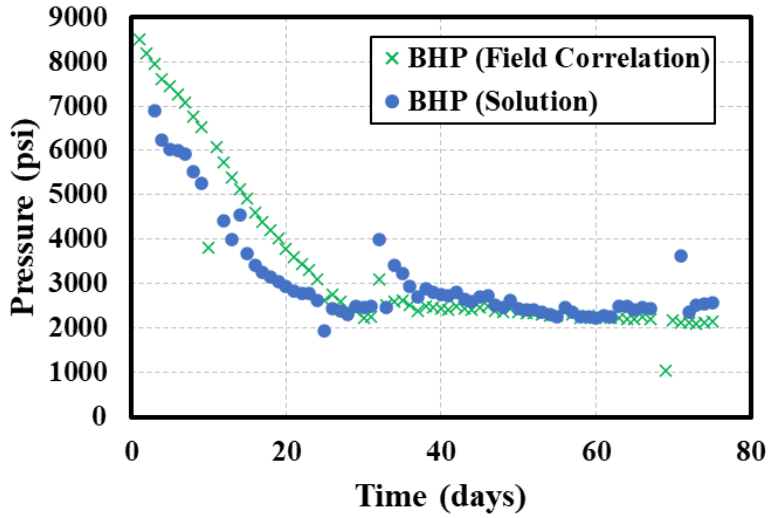

Figure 4.5: Comparing the bottom hole pressure (BHP) solution with the bottom hole pressure obtained from a field correlation.

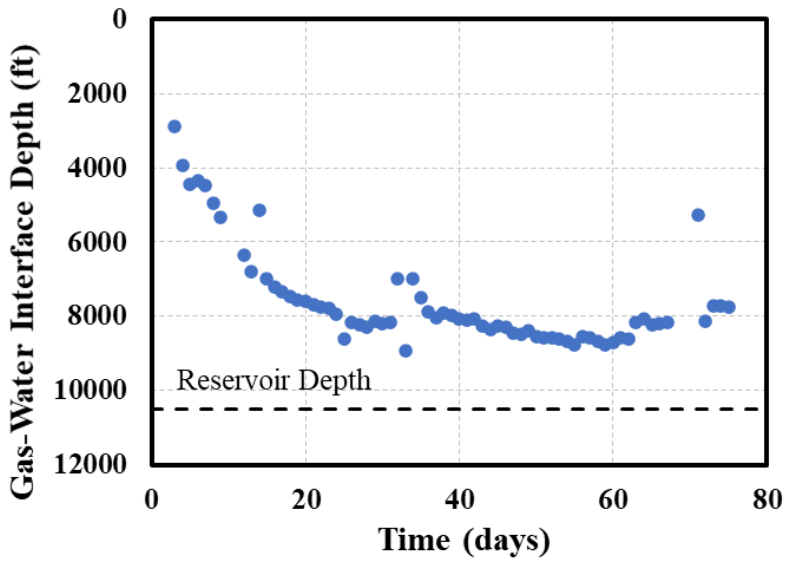

Figure 4.6: Changes in the gas-water interface depth as the water-gas ratio and wellhead pressure change.

Chapter 5: A Case Study in a Gas Condensate Reservoir

We have applied our workflow using microseismic events obtained from the field. The microseismic events were collected during the fracturing job of a horizontal well that has an 11280 ft horizontal section. The well was hydraulically fractured using 64 stages. Also, fluid production and pressure data were collected for 45 days.

In Chapter 3, we introduced a method to fit more than one fracture per stage. For this case study, we will explore two scenarios. In the first scenario, we will fit only one fracture per stage. While for the second scenario, we will allow the workflow to fit more than one fracture per stage. The number of fractures in each stage is controlled by DBSCAN, as explained in Chapter 3. The two scenarios will have identical results in the first two steps (stage definition and clustering), however, the results will be different starting from step 3 (fracture fitting).

5.1 Stage Definition

First, we define and separate the stages using the perforation intervals, as explained in Chapter 3. **Figure 5.1** shows the microseismic events before applying the stage definition step. Microseismic events from different stages are overlapping. The boundary lines that are generated using the perforation intervals are illustrated in **Figure 5.2**. We use the boundary lines to assign each microseismic event to the appropriate stage. **Figure 5.3** confirms that the overlapping issue is eliminated after applying the stage definition step. We use the actual perforation intervals from field data because the length of the perforation intervals may not be uniform, **Figure 5.4**.

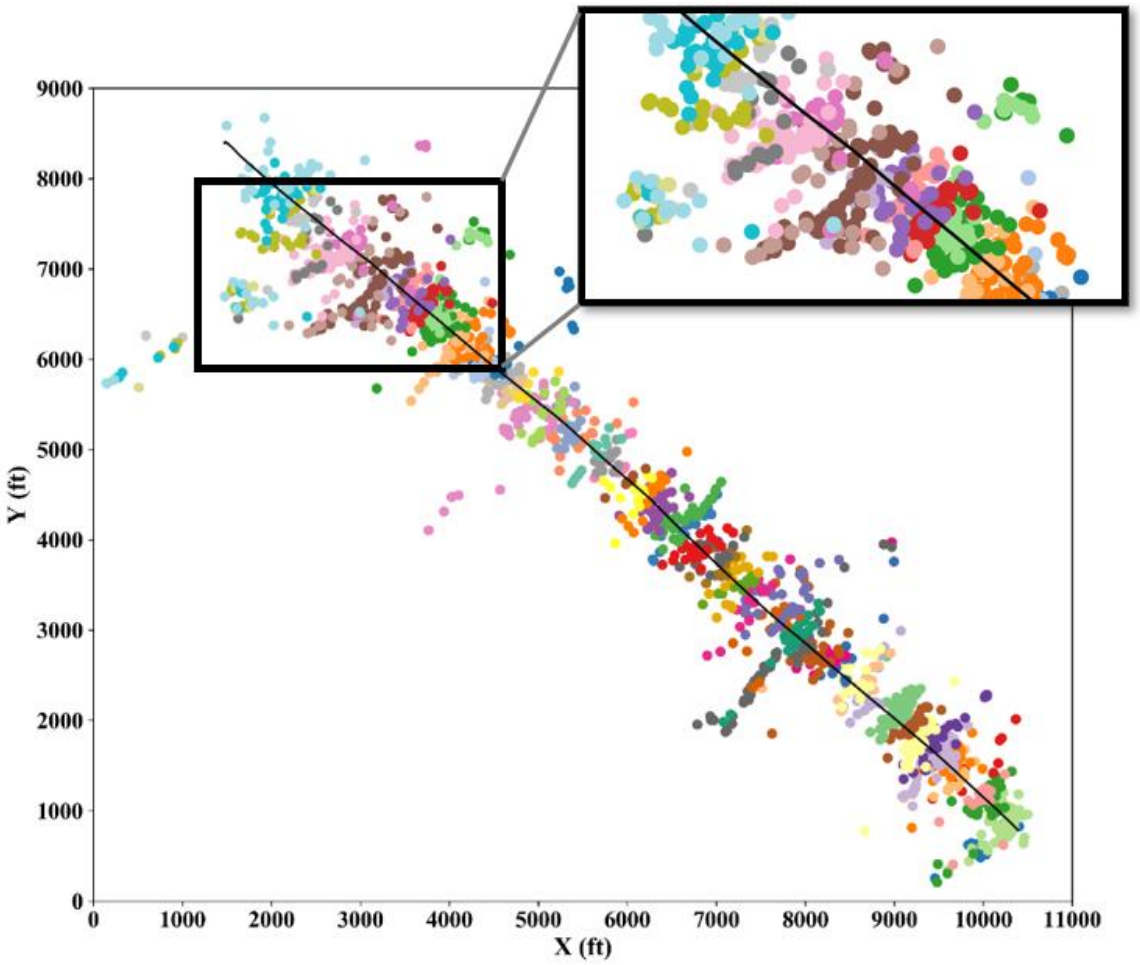

Figure 5.1: Original microseismic events. The zoomed cutout at the top-right corner highlights the overlapping between stages.

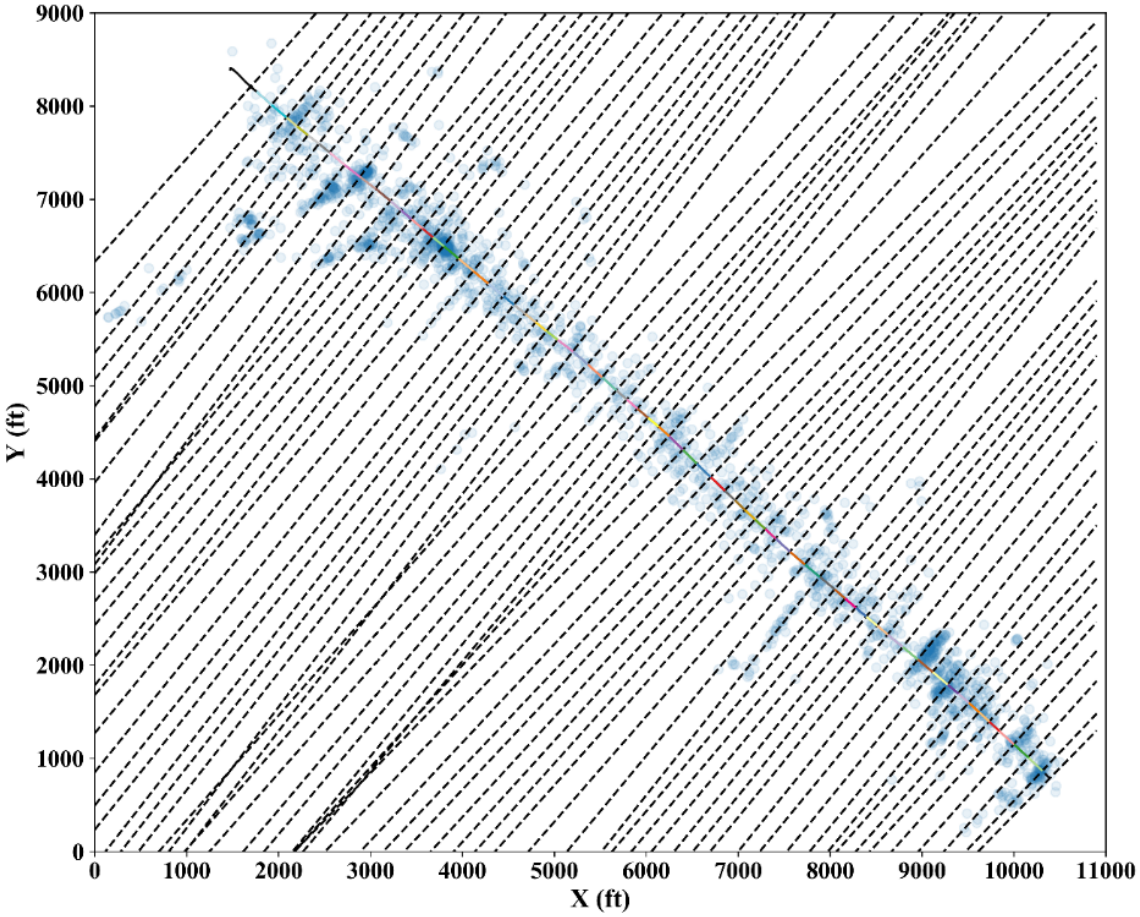

Figure 5.2: Using the perforation intervals to draw boundary lines that are perpendicular to the well path.

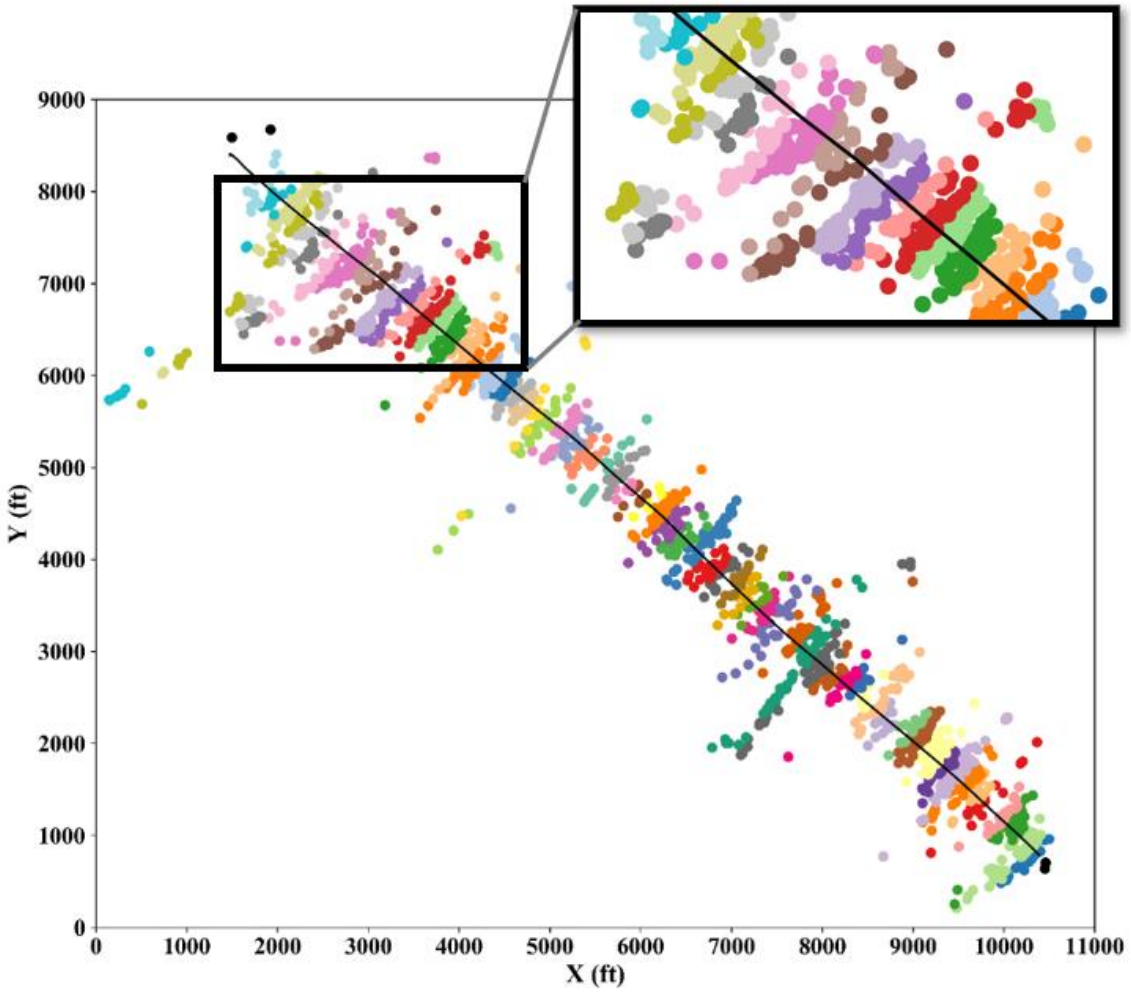

Figure 5.3: Modified microseismic events. Perforation intervals were used to define and separate the stages. The zoomed cutout at the top-right corner highlights that the overlapping issue between stages is resolved.

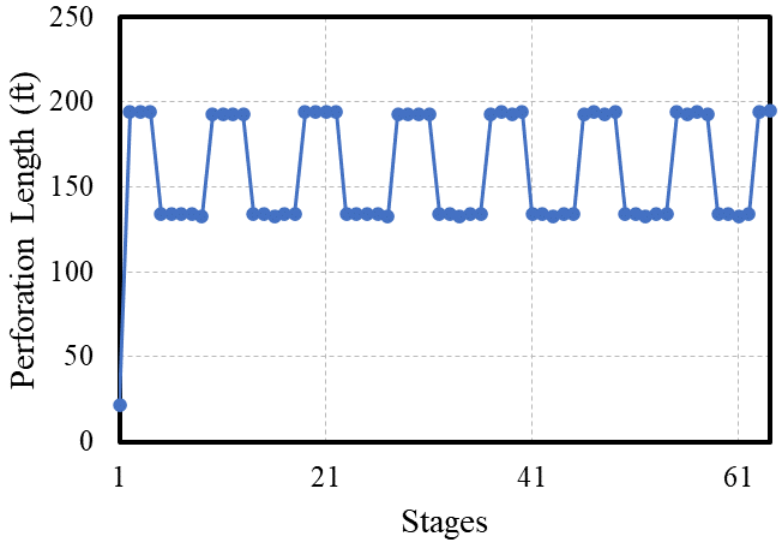

Figure 5.4: Different perforation length for each stage, as indicated from field data.

5.2 CLUSTERING

After defining and separating the stages, we ran DBSCAN for each stage to reduce the noise in the microseismic data and identify clusters if possible. **Figure 5.5** summarized the number of clusters found by DBSCAN in each stage. For the DBSCAN parameters, Eps was set to 250 ft and MinPts to 3. Any microseismic event identified as noise by DBSCAN is eliminated and is not included in any cluster. For the stages with more than one cluster, we choose one cluster that is closest to the wellbore.

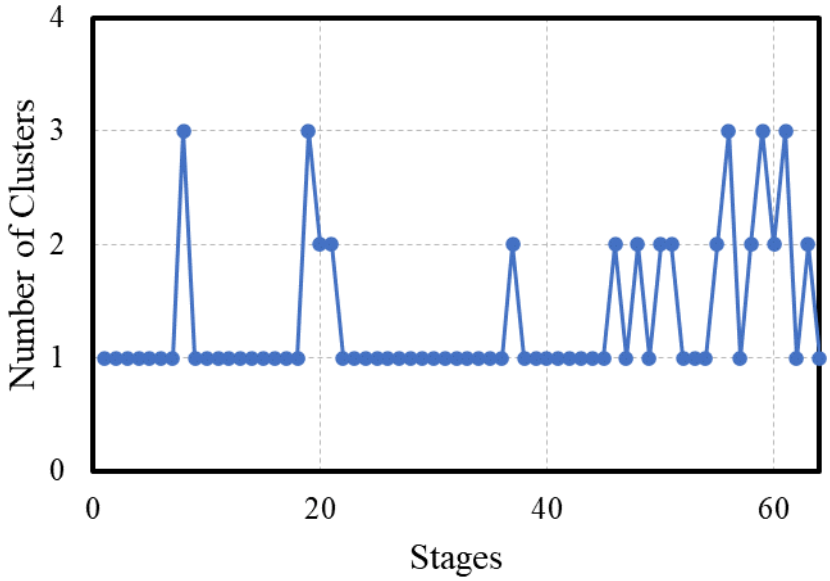

Figure 5.5: Number of clusters found by DBSCAN for each stage. Outliers are not included in any cluster. On average, DBSCAN found 1.3 clusters.

5.3 SCENARIO 1: FITTING ONE FRACTURE PER STAGE

5.3.1 Surface Fitting

In this section, we focus on the first scenario that assumes one fracture per stage. We used the surface fitting step to fit a fracture in the main cluster for each stage. We represent the fractures as polygons instead of rectangles. We use a maximum of 5 rectangles to build one polygon. **Figure 5.6** shows three fractures from three random stages. We use different colors for microseismic events to illustrate clusters.

After fitting the fractures, we calculate the fracture height, fractur half-length, dip angle, azimuth angle, and the coefficient of determination (R^2) for each fracture, **Figure 5.7** to **Figure 5.11**, respectively. We define the fracture height as the difference between the depth of the shallowest point in the polygon and the depth of the deepest point. In addition, we define the fracture half-length as the maximum horizontal distance between

any vertex in the polygon and the polygon's center. **Figure 5.12** presents the preliminary facture model. This model will be modified using scaling factors in the model calibration step.

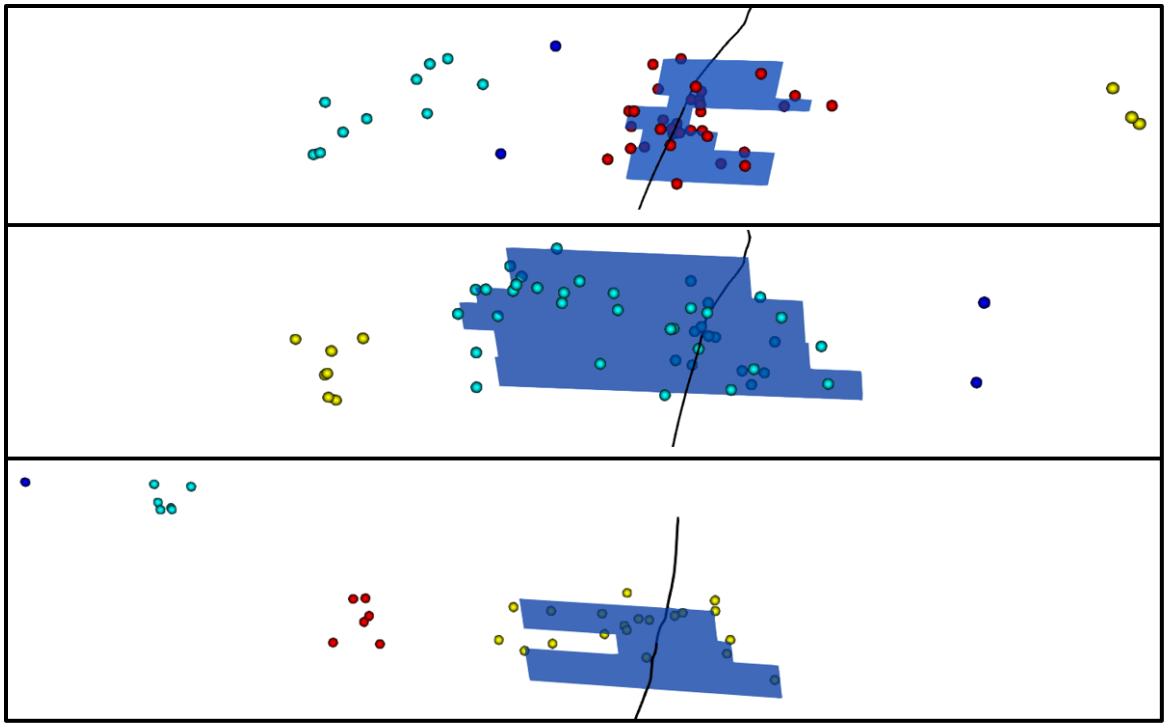

Figure 5.6: Fitting fracture planes in different stages. Three random stages were selected here for illustration.

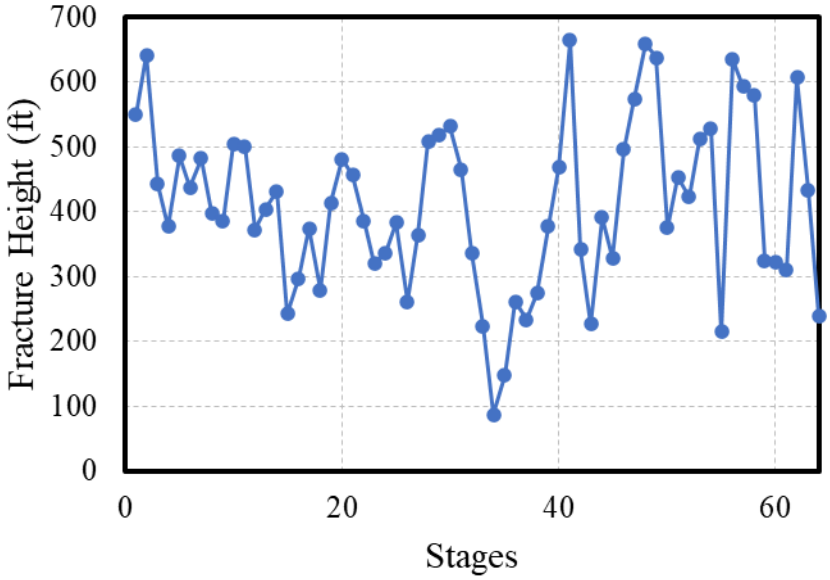

Figure 5.7: Fracture height variations in the preliminary fracture model. Average fracture height is 411 ft.

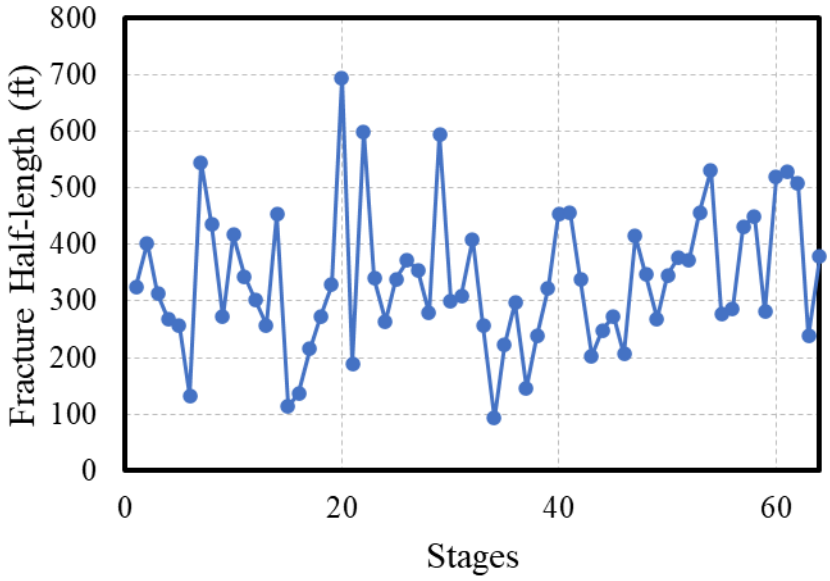

Figure 5.8: Fracture half-length variations in the preliminary fracture model. Average fracture half-length is 337 ft.

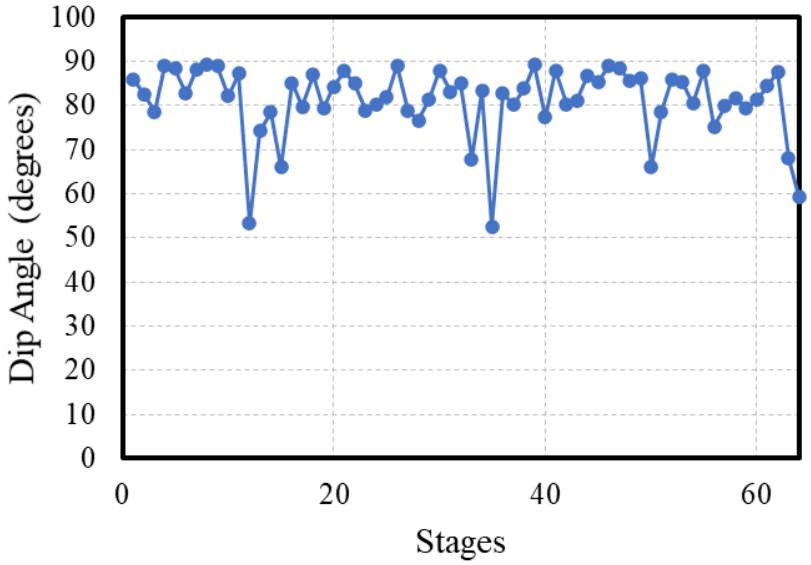

Figure 5.9: Dip angle variations in the fracture model. Average dip angle is 81 degrees.

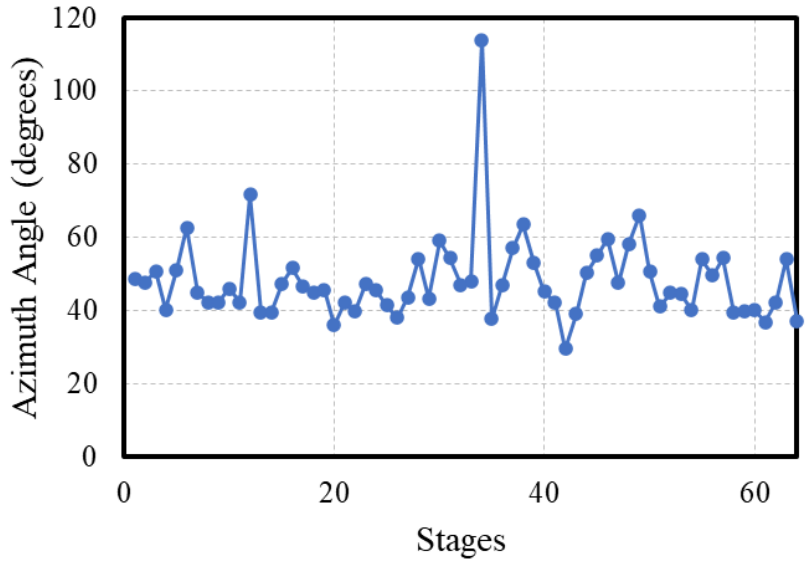

Figure 5.10: Azimuth angle variations in the fracture model. Average azimuth angle is 48 degrees.

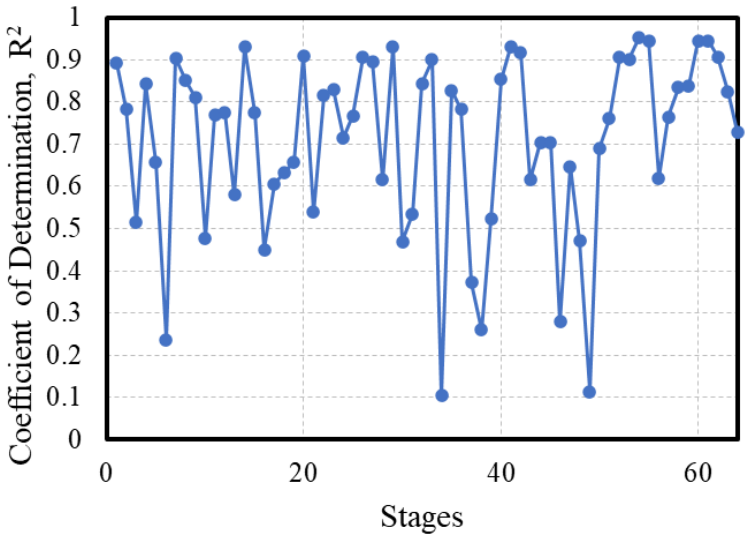

Figure 5.11: Coefficient of determination calculated for each fracture in the fracture model. Average coefficient of determination is 0.71.

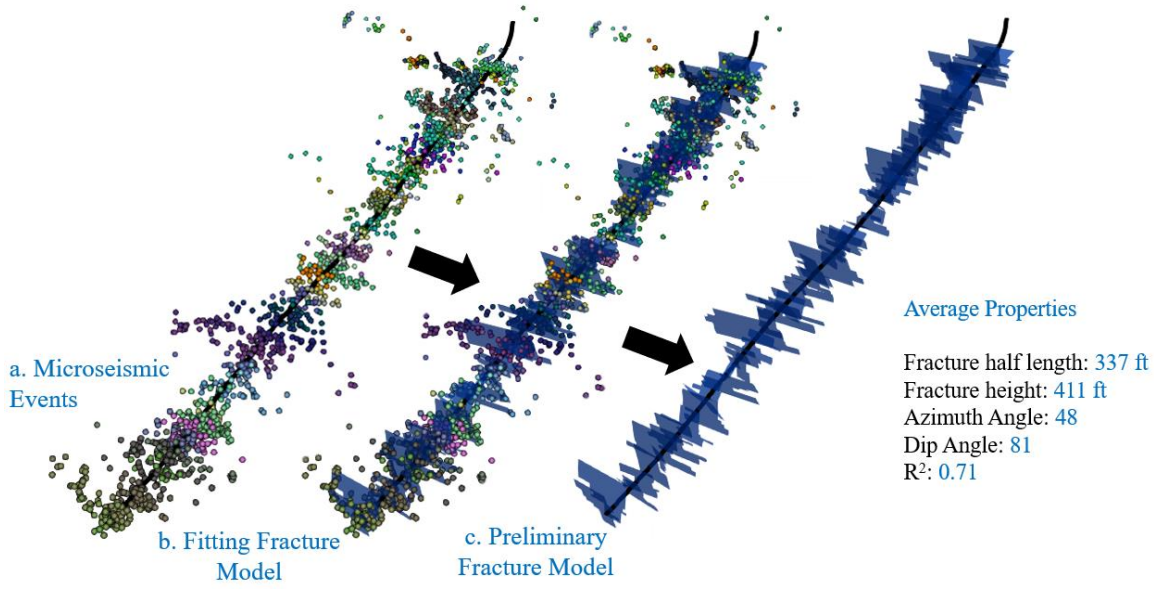

Figure 5.12: Preliminary fracture model. (a) Microseismic events after applying step 1. (b) Microseismic events and the preliminary fracture model are plotted together. (c) Preliminary fracture model (fracture model before applying the scaling factors).

5.3.2 Model Calibration

5.3.2.1 Reservoir Model

We built a simple reservoir simulation model to be used in the model calibration step. The main input parameters for the reservoir simulation model are summarized in **Table 5.1**. Also, **Table 5.2** provides some general information about the wellbore and fractures.

For the fluid model, we started with a compositional model from an offset well. However, the offset well was completed in a different formation and, hence, the gas oil ratio (GOR) was not matching with production data. Therefore, we calibrated the fluid model to match the average GOR from field data, **Figure 5.13**. The two-phase envelope for the final calibrated fluid model is presented in **Figure 5.14**. As suggested by the phase

envelope, at the initial reservoir condition, only gas is present in the reservoir. Condensate is separated from solution only at the wellbore conditions. Also, since the field GOR is relatively constant at around 3452 scf/STB, we expect the pressure around the wellbore to remain above the dew point pressure during the simulation period.

Table 5.1: Basic parameters used to build the reservoir simulation model.

Parameter	**Value**	**Unit**
Model dimension (x × y × z)	12000 × 3000 × 850	ft
Number of gird blocks (x × y × z)	300 × 30 × 5	-
Grid block dimension (x × y × z)	40 × 100 × 170	ft
Initial reservoir pressure	5280	psi
Reservoir temperature	230	°F
Porosity	0.06	-
Permeability	118	Nano Darcy
Initial water saturation	0.34	-

Table 5.2: General information about the horizontal well and hydraulic fractures.

Parameter	**Value**	**Unit**
Horizontal well length	11280	ft
Number of stages	64	-
Number of fractures	64	-
Fracture spacing	57-304	ft

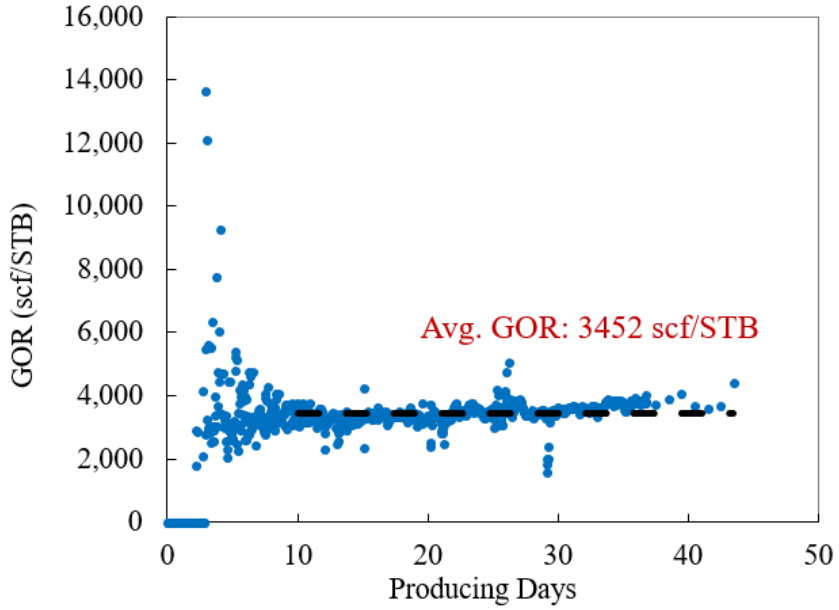

Figure 5.13: Gas oil ratio (GOR) history from field production data. GOR is relatively constant with an average GOR of 3452 scf/STB.

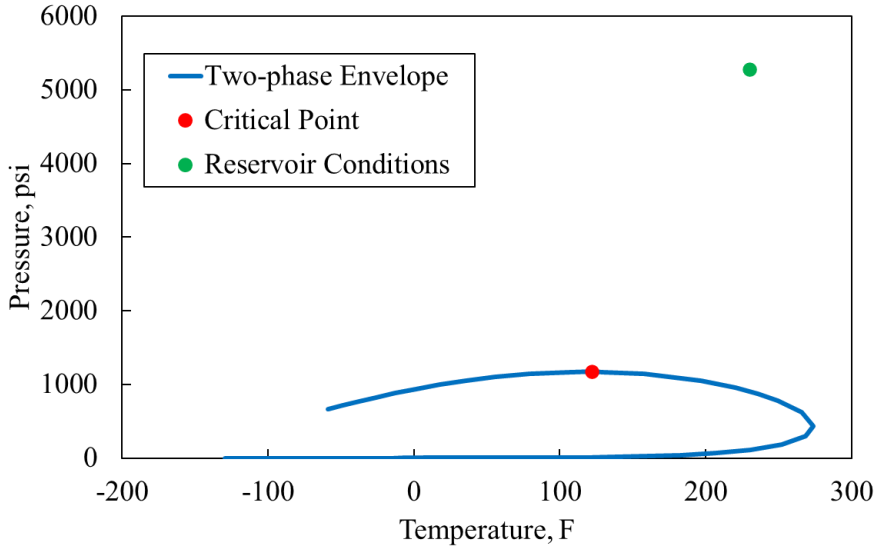

Figure 5.14: Two-phase envelop used in the reservoir simulation model. At the initial reservoir conditions, only gas is present in the reservoir.

5.3.2.2 Assisted History Matching

We used the assisted history matching (AHM) workflow described in Chapter 3 to calibrate our fracture model. We identified five uncertain parameters to be considered in the AHM workflow. In addition to the two scaling factors for fracture height and fracture half-length, we include fracture conductivity, fracture width, and fracture water saturation as uncertain parameters. The minimum and maximum values for these uncertain parameters are specified in **Table 5.3**.

Table 5.3: Uncertain parameters selected to calibrate the preliminary fracture model in the AHM workflow.

Parameter	**Min**	**Max**	**Unit**
Fracture half-length scaling Factor	0.1	1	-
Fracture height scaling factor	0.1	1	-
Fracture conductivity	80	100	md-ft
Fracture width	0.01	0.5	ft
Fracture water saturation	0.8	0.9	-

The AHM workflow has identified 53 history matching solutions. **Figure 5.15** shows the weighted root mean square error (RMSE) for all simulation runs. The AHM workflow was finding history matching solutions and exploring the uncertainty domain consistently throughout the iterations.

The daily and cumulative production results for the 53 history matching solutions are illustrated in **Figure 5.16** and **Figure 5.17**, respectively. The gas production rate and cumulative production is matching the field data perfectly because we used the gas rates to constrain the well's production. Also, all the history matching solutions have identical oil

production rates because the pressure around the wellbore was always higher than the dew point pressure during the simulation period.

In addition, we can obtain a better estimate of the uncertain parameters by reviewing the distribution of uncertain parameters in the history matching solutions, **Figure 5.18**. For example, we considered the scaling factors in the range from 0.1 to 1. However, all the history matching solutions have scaling factors above 0.4. Therefore, we have narrowed down the uncertainty range in the scaling factor parameters.

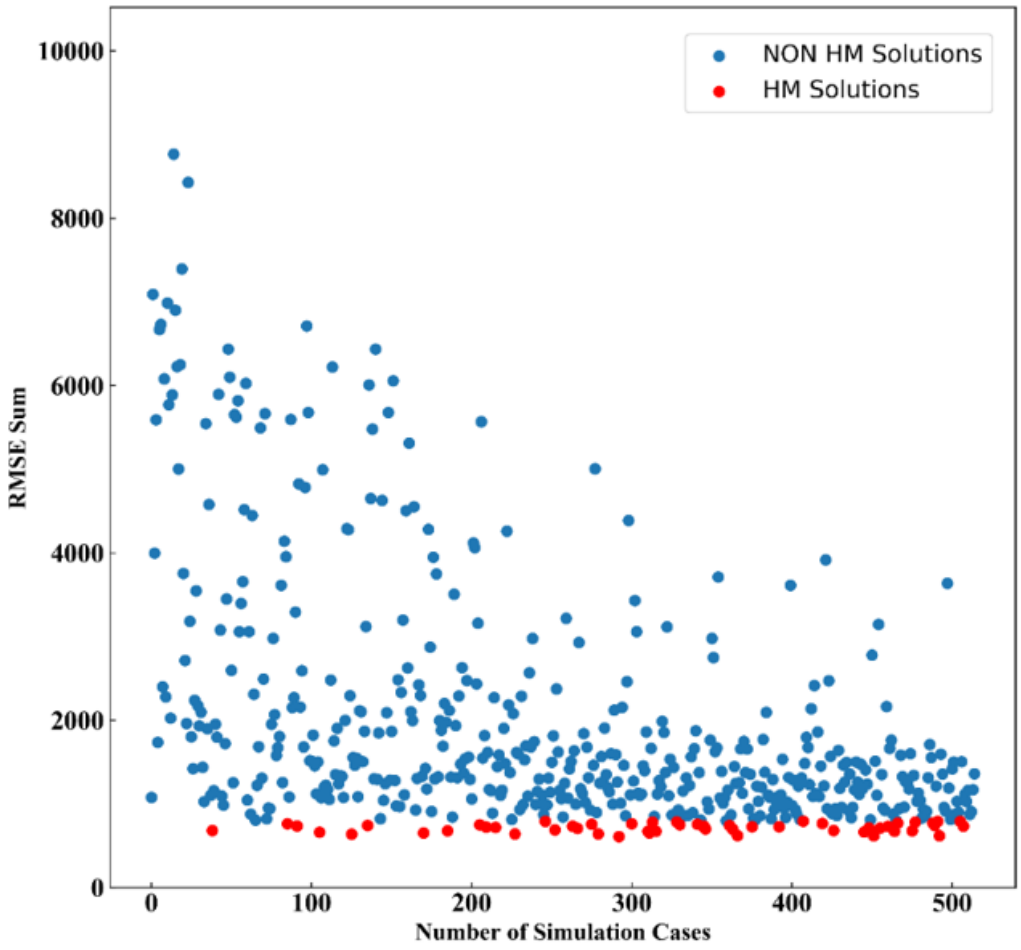

Figure 5.15: Weighted root mean square error (RMSE) for all 515 simulation runs.

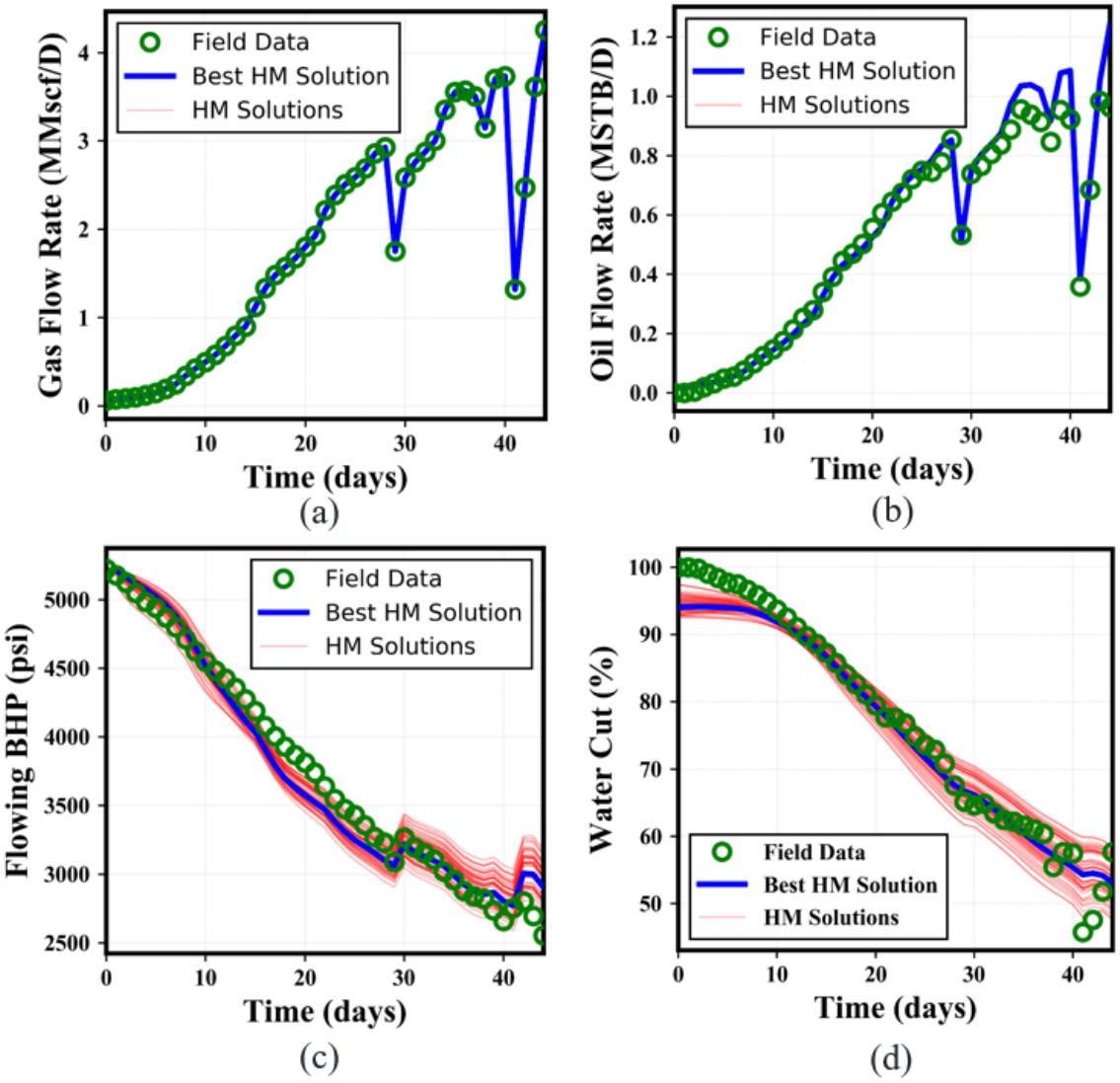

Figure 5.16: Daily production and pressure results from the history matching solutions compared to field data. (a) Gas flow rate. (b) Oil flow rate. (c) Flowing bottom hole pressure (BHP). (d) Water cut.

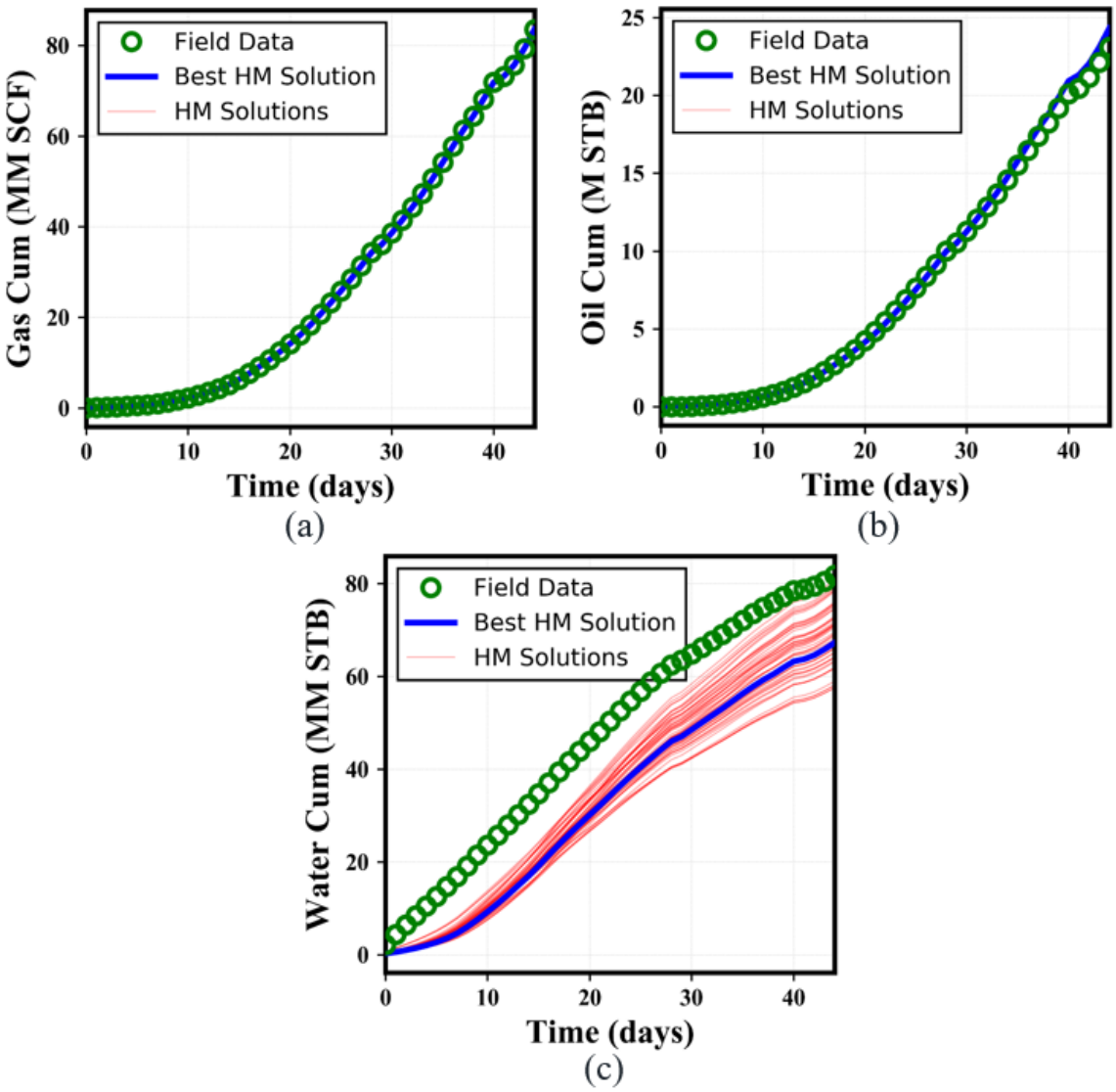

Figure 5.17: Cumulative production data from the history matching solutions compared to field data. (a) Cumulative gas production. (b) Cumulative oil production. (c) Cumulative water production.

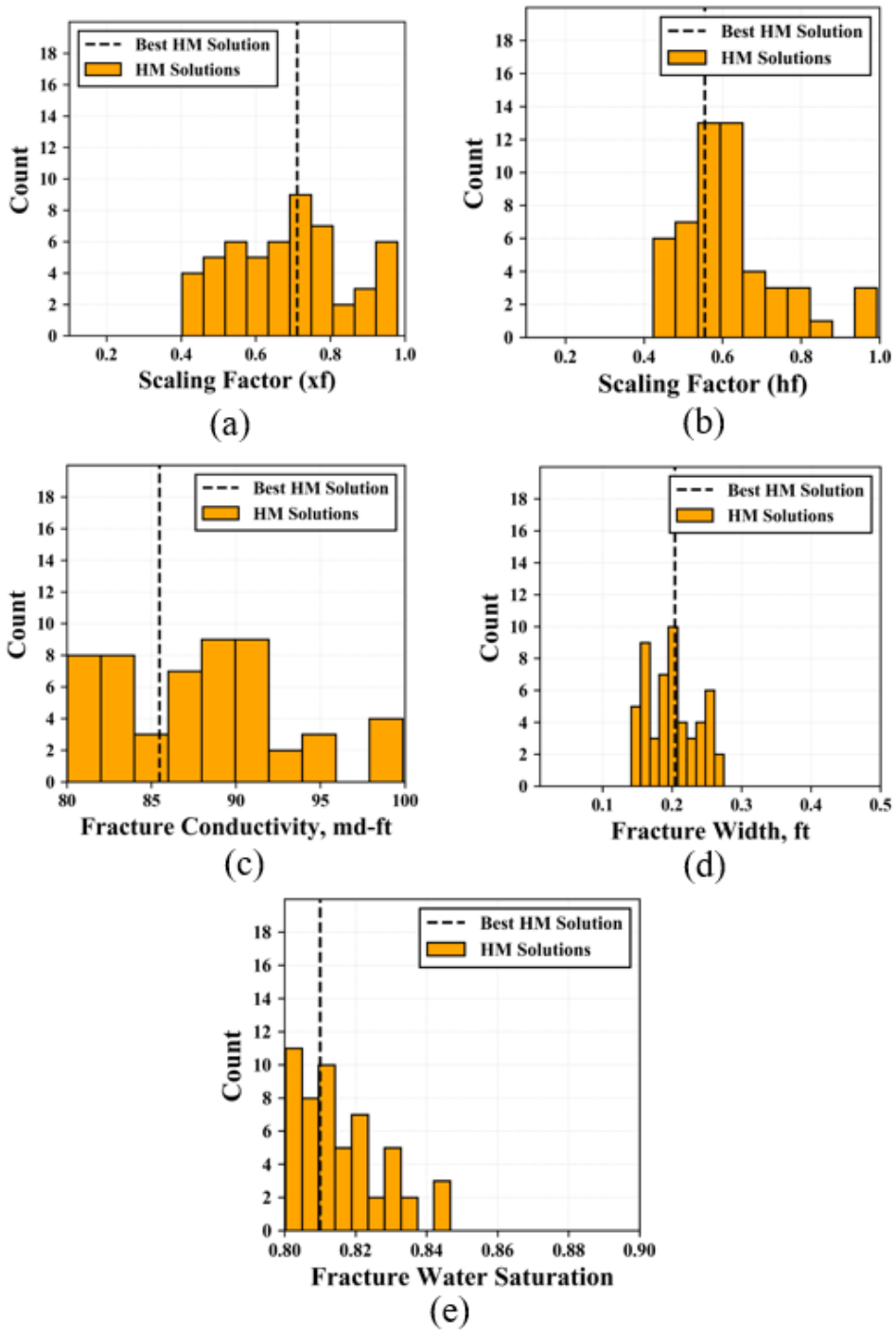

Figure 5.18: Distribution of uncertain parameters in the history matching solutions. (a) Scaling factor for fracture half-length. (b) Scaling factor for fracture height. (c) Fracture conductivity. (d) Fracture width. (e) Fracture water saturation.

Using the scaling factors from the history matching solution, we can generate several versions of the fracture model. We picked the scaling factors from the best history matching solution (the one with the lowest RMSE), to modify the fracture model. The best history matching solution is represented by dashed black lines in Figure 5.18 and blue curves in Figure 5.16 and Figure 5.17. The scaling factors for the best history matching solutions are 0.55 and 0.71 for fracture height and fracture half-length, respectively. We used these scaling factors to modify the fracture height and fracture half-length in the preliminary fracture model, **Figure 5.19** and **Figure 5.20**, respectively. The new calibrated fracture model is presented in **Figure 5.21**. The whole process we used to obtain the final calibrated fracture model is also illustrated in Figure 5.21.

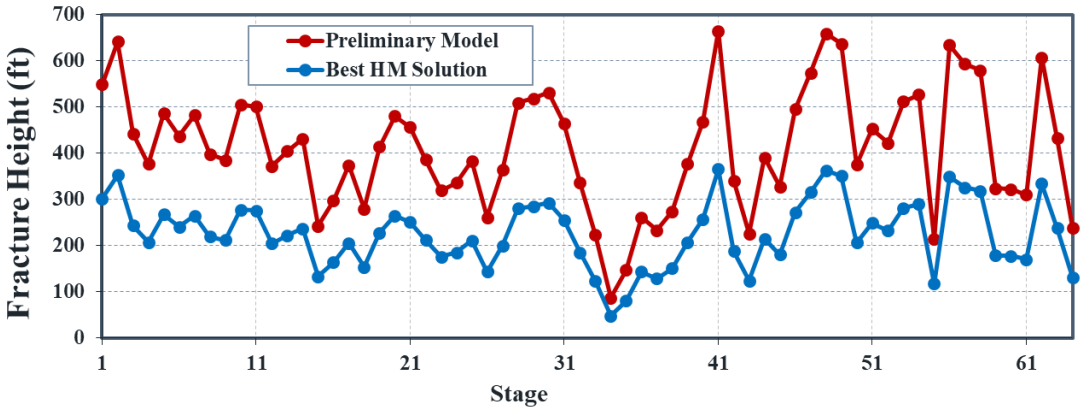

Figure 5.19: Comparing the fracture height in the preliminary fracture model with the fracture height in the calibrated fracture model (best HM solution).

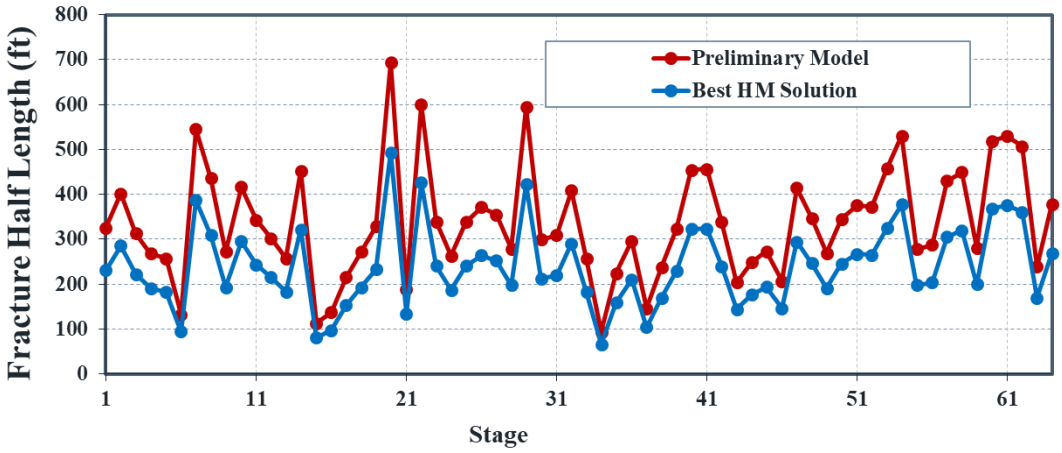

Figure 5.20: Comparing the fracture half-length in the preliminary fracture model with the fracture half-length in the calibrated fracture model (best HM solution).

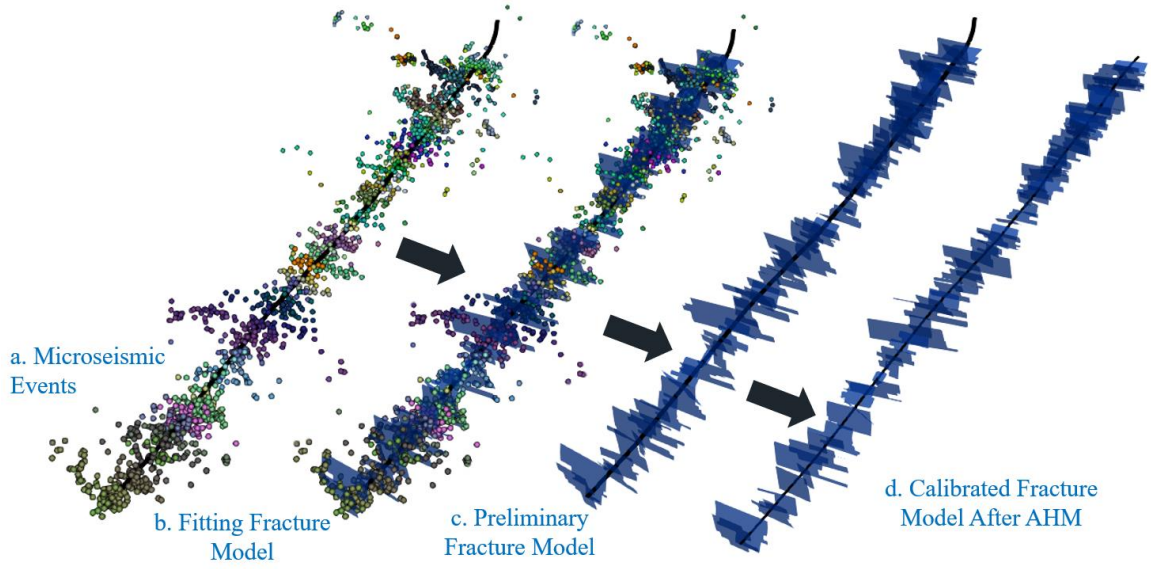

Figure 5.21: Overall process to find the calibrated fracture model. (a) Microseismic events after applying step 1. (b) Microseismic events and the preliminary fracture model are plotted together. (c) Preliminary fracture model (fracture model before applying the scaling factors). (d) Final fracture model after applying the scaling factors from the best HM solution.

5.4 SCENARIO 2: FITTING MULTIPLE FRACTURES PER STAGE

5.4.1 Surface Fitting

In this scenario, we fit more than one fracture per stage using an additional step described in Chapter 3. After the initial fracture fit, we apply DBSCAN to the error distribution to find clusters. Then, for each cluster, we fit a new fracture. **Figure 5.22** presents some examples for fitting multiple fractures in a single stage. We rely on DBSCAN to identify the number of clusters in each stage. Therefore, the number of fractures varies for each stage, **Figure 5.23**. In this case, we were able to fit 111 fractures in the 64 stages using 15 ft and 2 for the DBSCAN parameters Eps and MinPts, respectively. Our average cluster efficiency is 14% considering 12 perforations per stage, **Figure 5.24**.

The fracture height, fracture half-length, dip angle, azimuth angle, coefficient of determinations for all 111 fractures are illustrated in **Figure 5.25** to **Figure 5.29**, respectively. Average coefficient of determination (R^2) is 0.93, compared to 0.71 for Scenario 1. We were able to achieve higher R^2 because the number of microseismic events used to fit a fracture is smaller in Scenario 2. Achieving higher R^2 with smaller dataset is easier than with larger dataset. The average height and half-length is smaller in Scenario 2, which is expected because in Scenario 2 we divide the dataset into smaller subsets. The average dip and azimuth are the same as Scenario 1. The preliminary fracture model for Scenario 2 is presented in **Figure 5.30**.

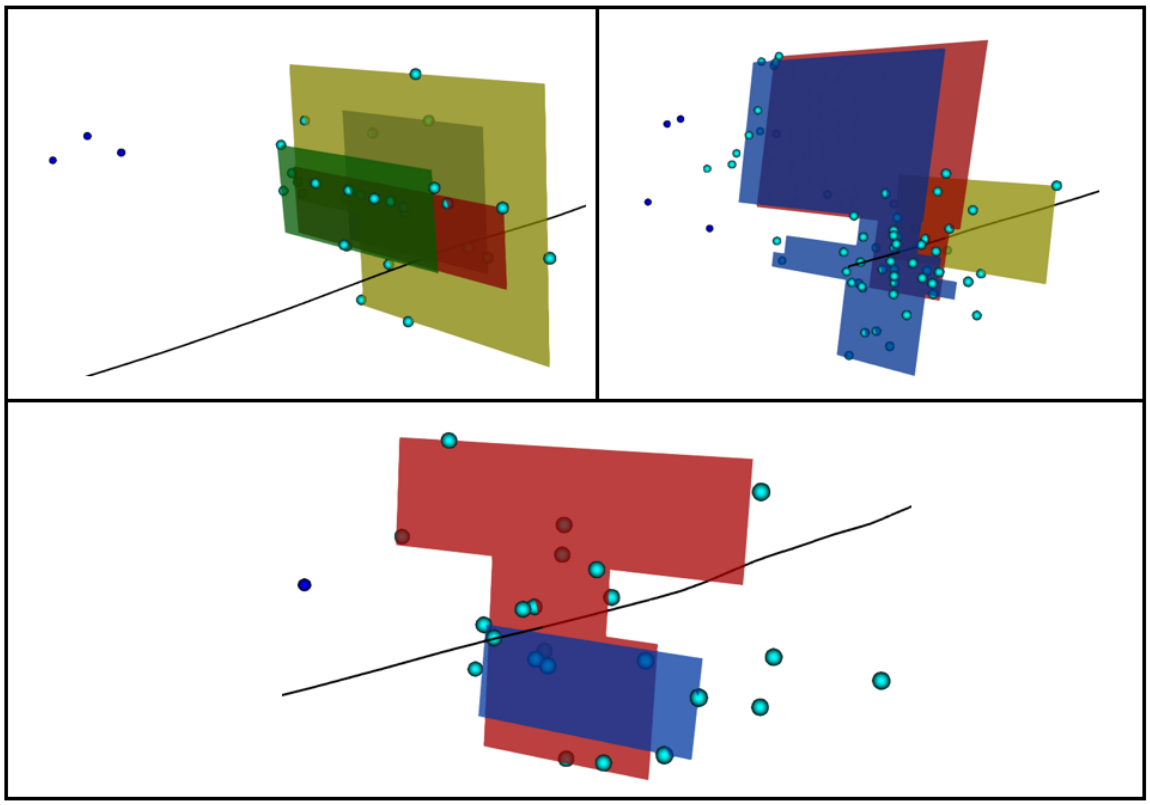

Figure 5.22: Fitting multiple fracture planes in different stages. Three random stages were selected here for illustration. DBSCAN parameters (Eps=15 ft, MinPts=2).

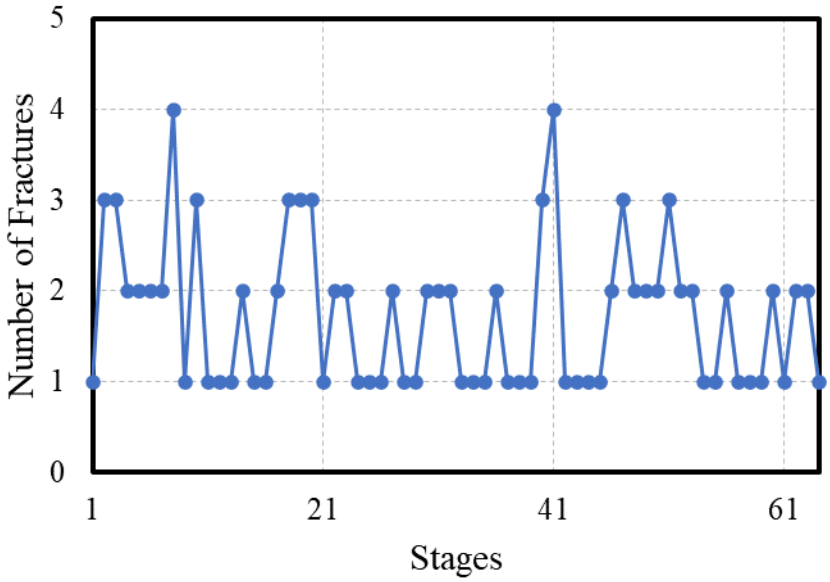

Figure 5.23: Number of fractures fitted for each stage. A total of 111 fractures were fitted in 64 stages. DBSCAN parameters (Eps=15 ft, MinPts=2).

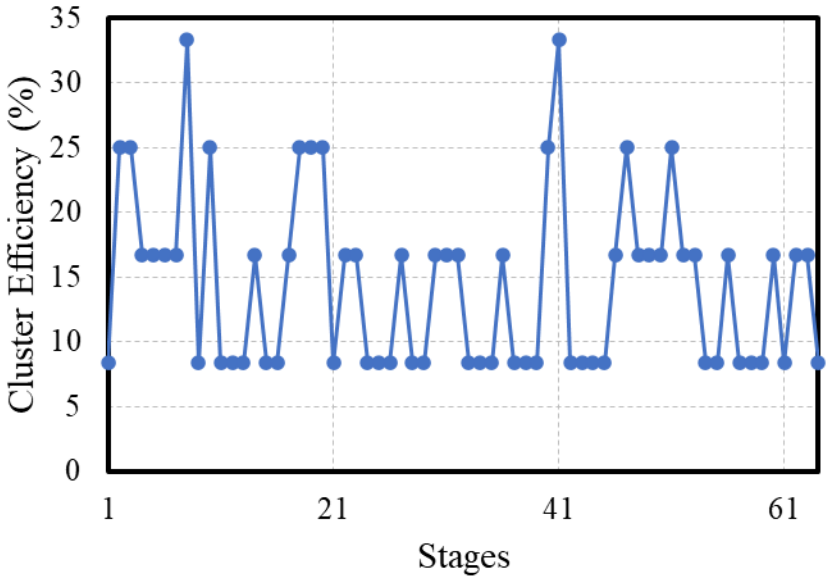

Figure 5.24: Cluster efficiency for each stage considering 12 perforations per stage. Average cluster efficiency is 14%.

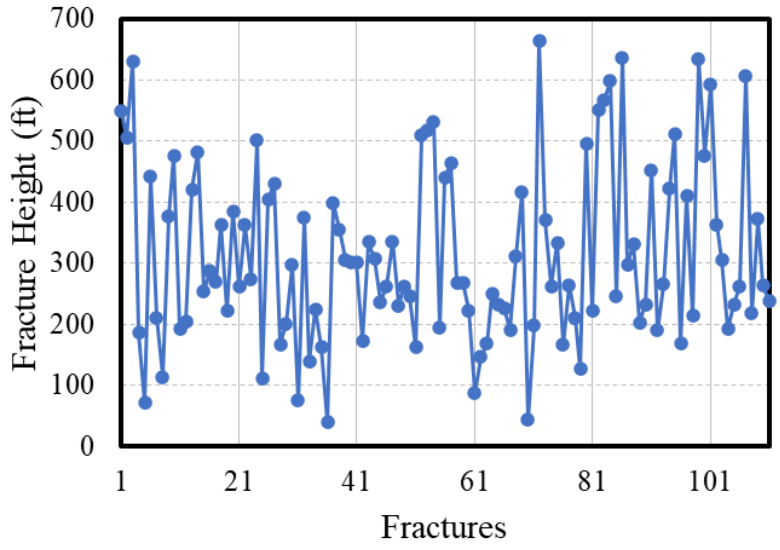

Figure 5.25: Fracture height variations in the preliminary fracture model. Average fracture height is 313 ft.

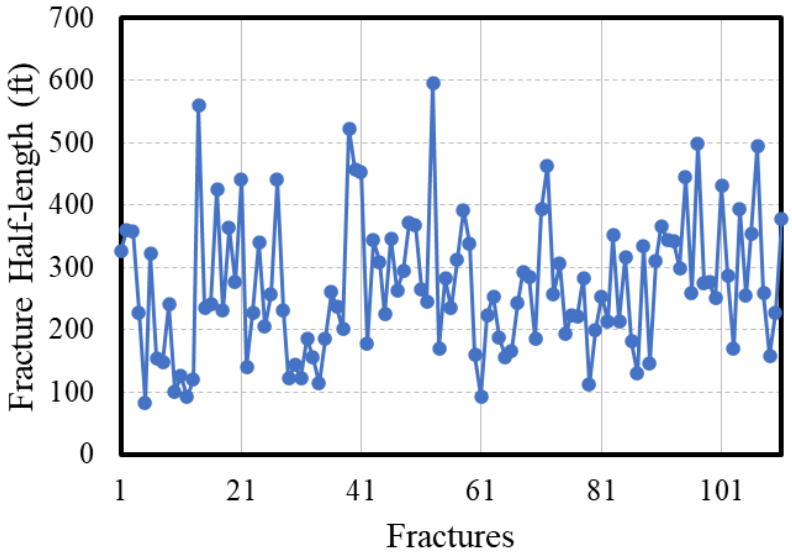

Figure 5.26: Fracture half-length variations in the preliminary fracture model. Average fracture half-length is 270 ft.

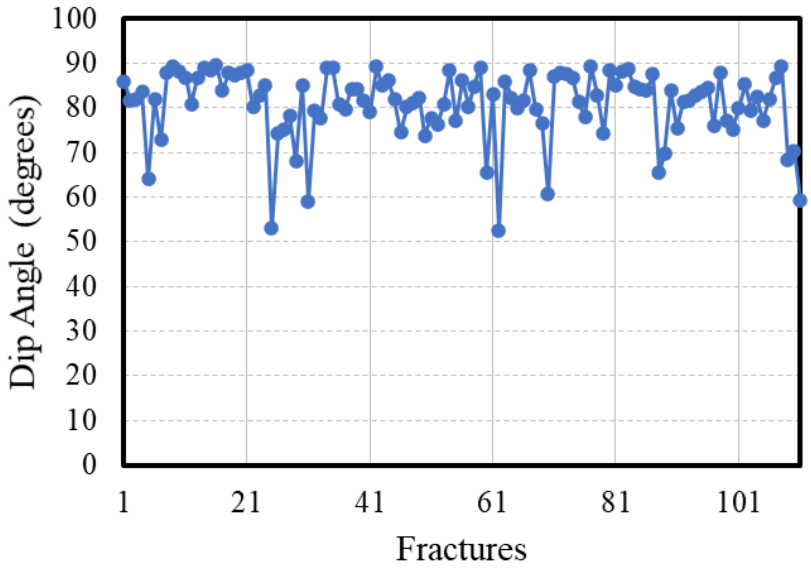

Figure 5.27: Dip angle variations in the fracture model. Average dip angle is 81 degrees.

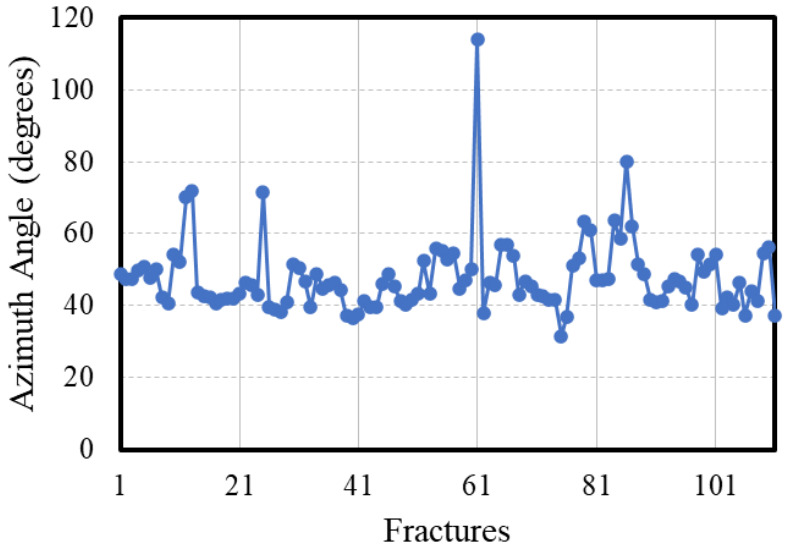

Figure 5.28: Azimuth angel variations in the fracture model. Average azimuth angle is 48 degrees.

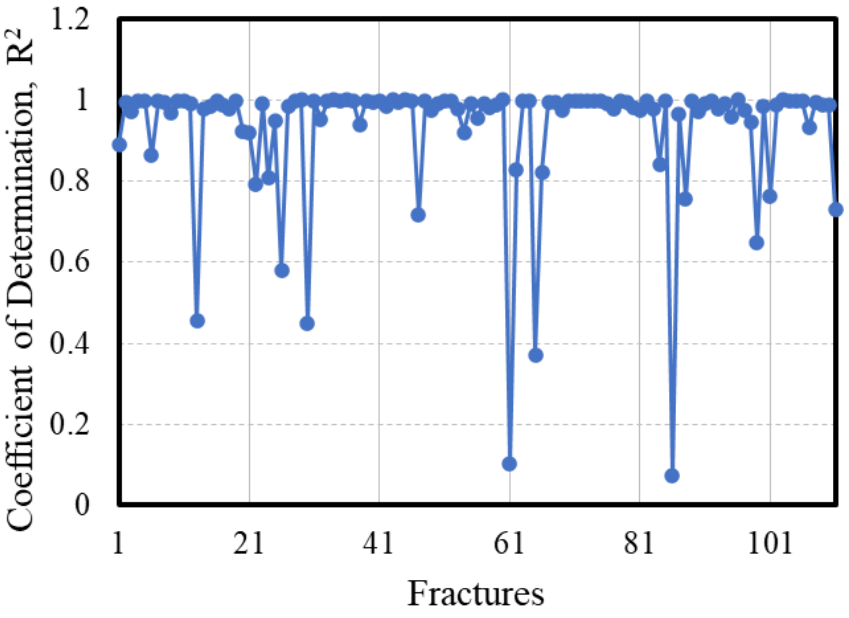

Figure 5.29: Coefficient of determination calculated for each fracture in the fracture model. Average coefficient of determination is 0.93.

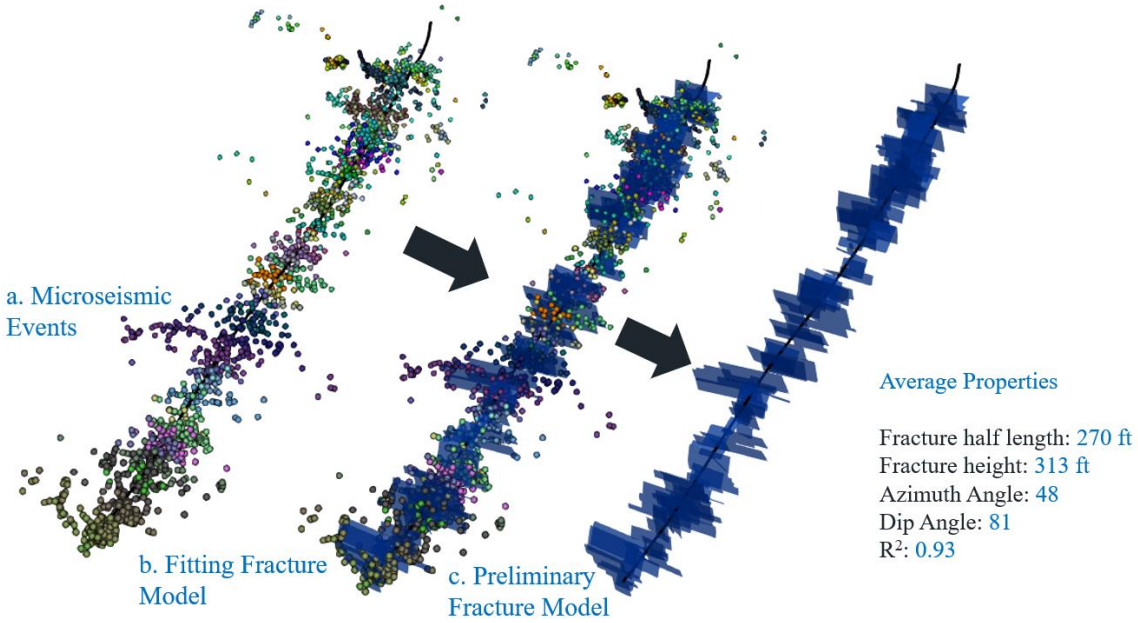

Figure 5.30: Preliminary fracture model. (a) Microseismic events after applying step 1. (b) Microseismic events and the preliminary fracture model are plotted together. (c) Preliminary fracture model (fracture model after applying the scaling factors).

5.4.2 Model Calibration

5.4.2.1 Reservoir Model

We used the same reservoir model as in Scenario 1. The only difference in Scenario 2 is the fracture model here has more fractures. **Table 5.4** shows some general information about the fracture model.

Table 5.4: General information about the hydraulic fractures.

Parameter	**Value**	**Unit**
Number of stages	64	-
Number of fractures	111	-
Fracture spacing	7-345	ft

5.4.2.2 Assisted History Matching

We used similar uncertain parameters for Scenario 2, **Table 5.5**. We changed the minimum value for the scaling factors from 0.1 to 0.2 to reduce the size of the uncertainty domain and find more history matching solutions.

Table 5.5: Uncertain parameters selected to calibrate the preliminary fracture model in the AHM workflow.

Parameter	**Min**	**Max**	**Unit**
Fracture half-length scaling Factor	0.2	1	-
Fracture height scaling factor	0.2	1	-
Fracture conductivity	80	100	md-ft
Fracture width	0.01	0.5	ft
Fracture water saturation	0.8	0.9	-

Using the AHM workflow, a total of 62 history matching solutions were found, **Figure 5.31**. The history matching solutions were chosen based on their weighted root mean square error (RMSE). **Figure 5.32** and **Figure 5.33** show the daily and cumulative production results in the history matching solution, respectively. The gas production rate was selected to constrain the well's production.

In addition, we reviewed the distribution of uncertain parameters for the history matching solutions, **Figure 5.34**. In Scenario 1, the scaling factors for the history matching solutions are all above 0.4. However, in Scenario 2, we have found some solutions with scaling factors less than 0.4. This result is expected because we are using more fractures in Scenario 2 and, hence, smaller fractures are expected to match the same production data.

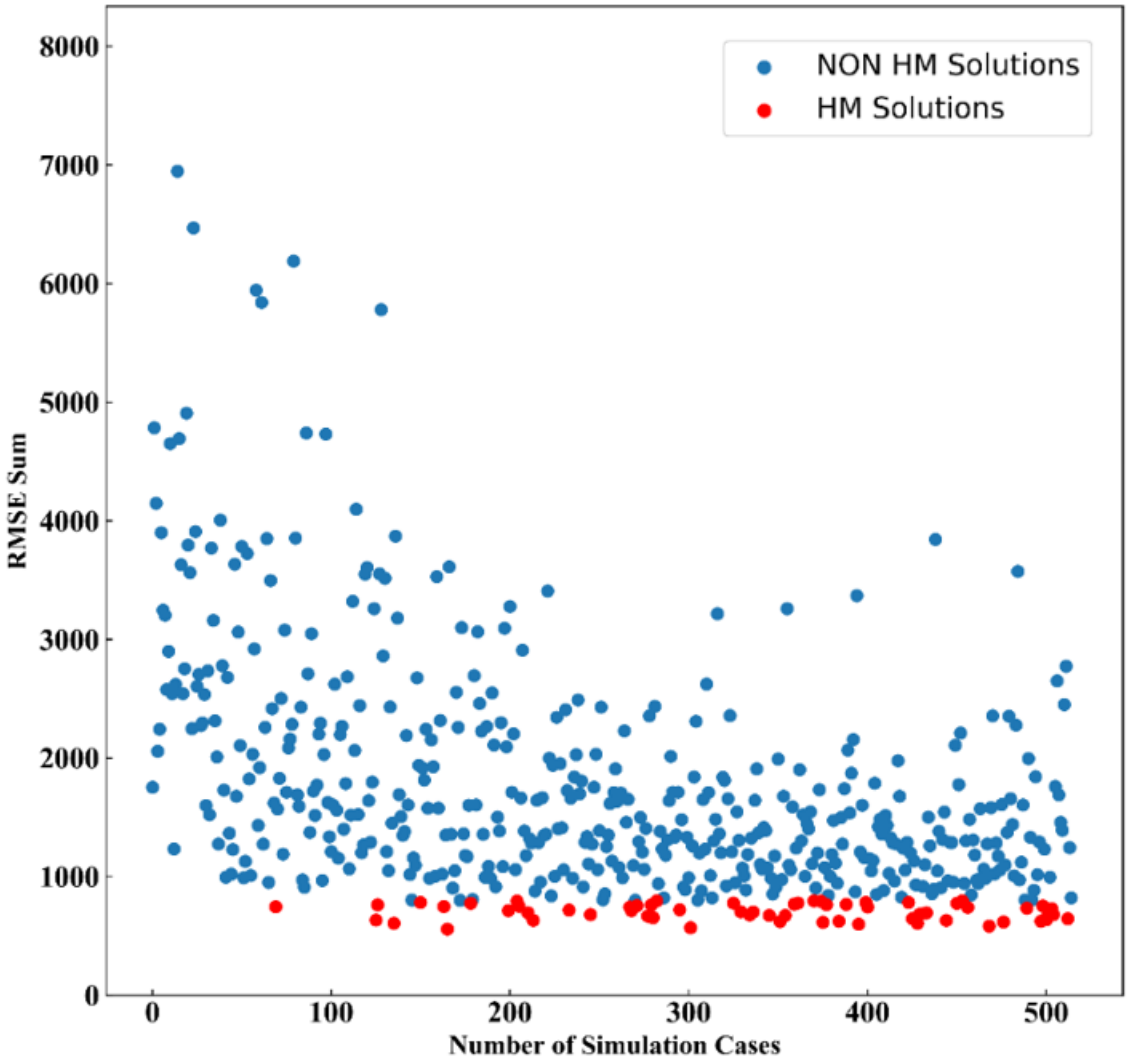

Figure 5.31: Weighted root mean square error (RMSE) for all 515 simulation runs.

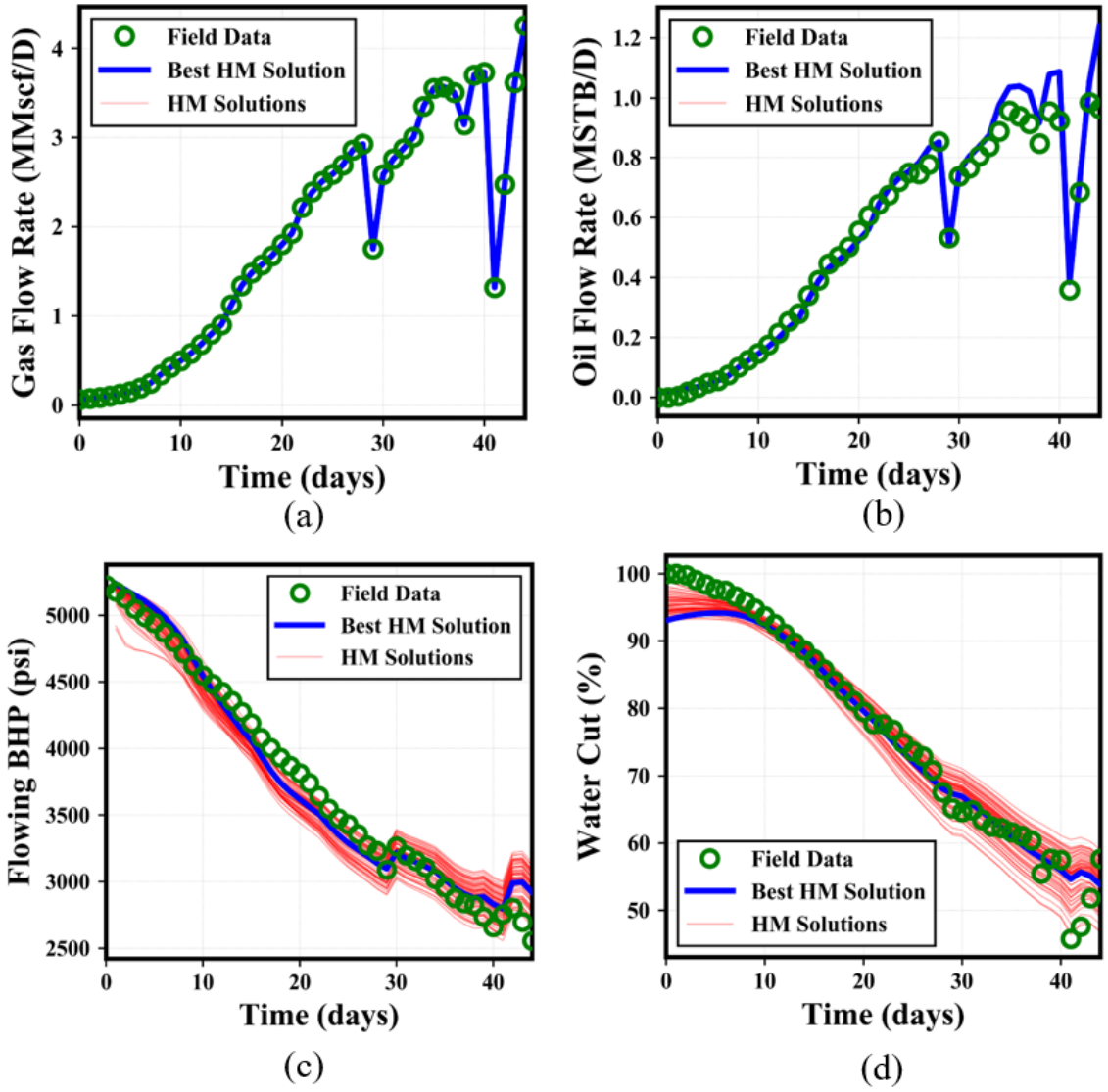

Figure 5.32: Daily production and pressure results from the history matching solutions compared to field data. (a) Gas flow rate. (b) Oil flow rate. (c) Flowing bottom hole pressure (BHP). (d) Water cut.

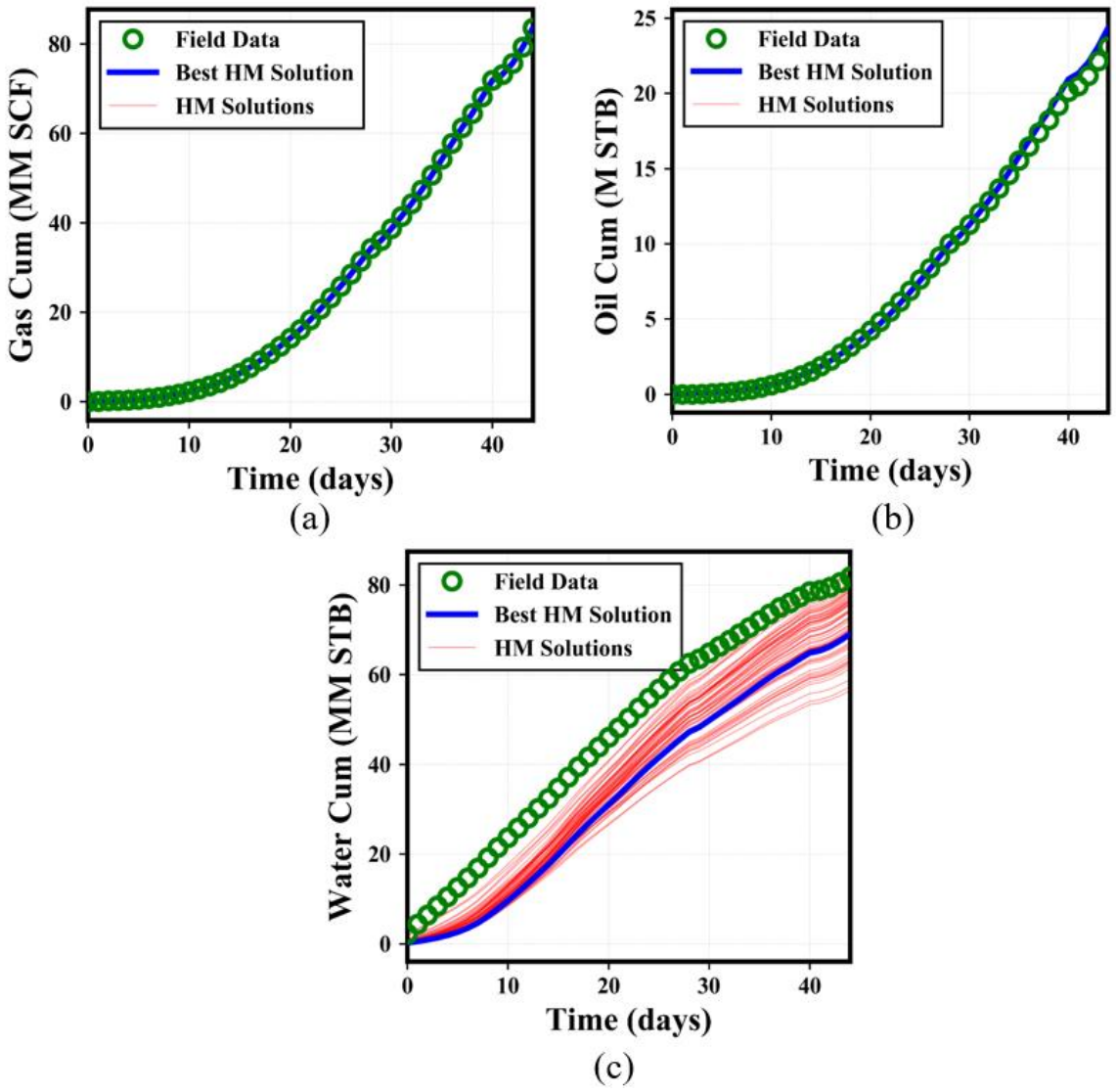

Figure 5.33: Cumulative production data from the history matching solutions compared to field data. (a) Cumulative gas production. (b) Cumulative oil production. (c) Cumulative water production.

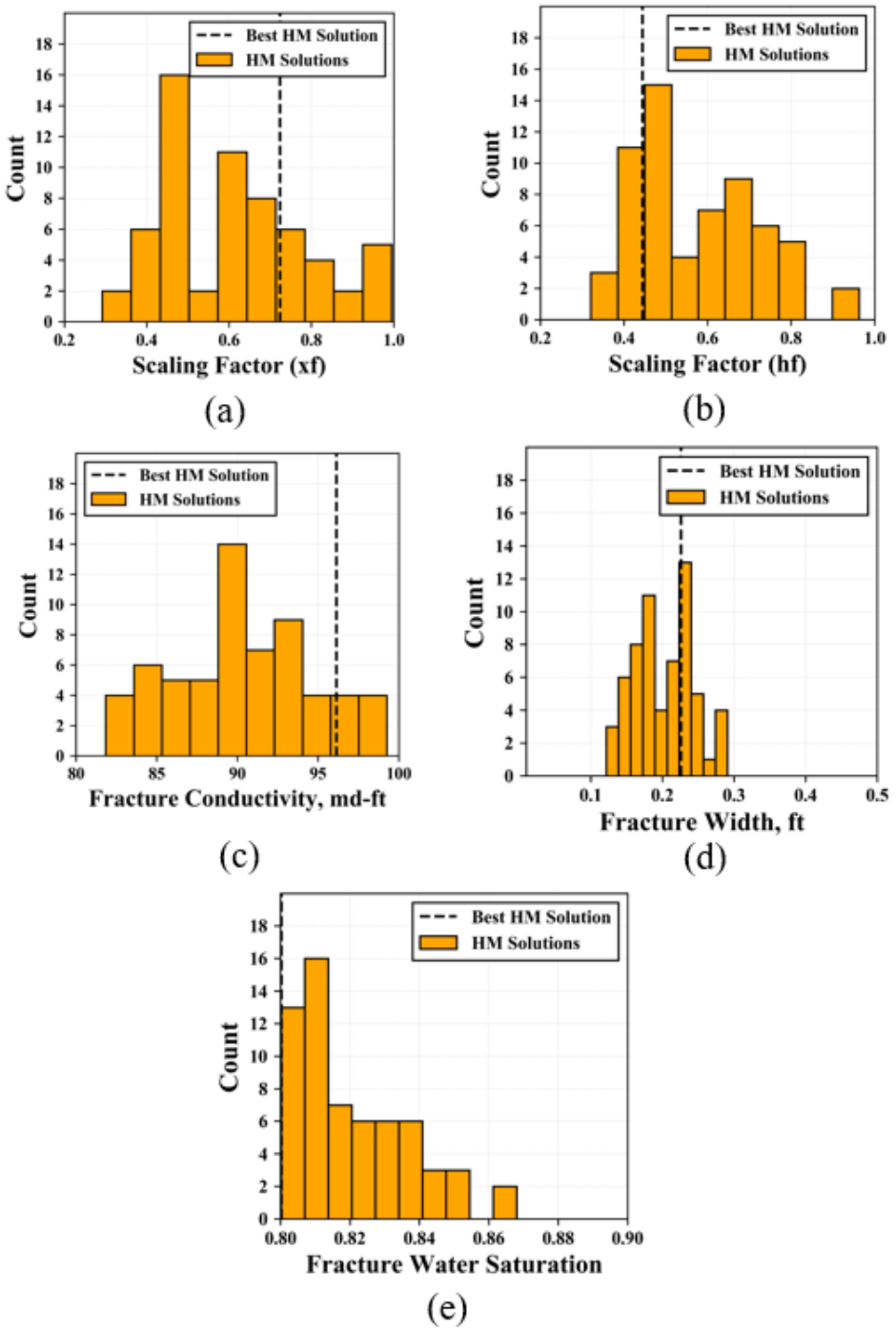

Figure 5.34: Distribution of uncertain parameters in the history matching solutions. (a) Scaling factor for fracture half-length. (b) Scaling factor for fracture height. (c) Fracture conductivity. (d) Fracture width. (e) Fracture water saturation.

Using the scaling factors for the best history matching (HM) solution, we modified the fracture height and fracture half-length in our model, **Figure 5.35** and **Figure 5.36**, respectively. In the best HM solution, the scaling factors are 0.44 and 0.72 for the fracture height and fracture half-length, respectively. Both scenarios provide similar scaling factors for fracture half-length in the best HM solution. However, the scaling factor for fracture height is smaller in Scenario 2. Therefore, as we add more fractures to the model, smaller area is needed to match the same production data. Finally, **Figure 5.37** includes the final calibrated fracture model and summarized the entire process we used to obtain the fracture model.

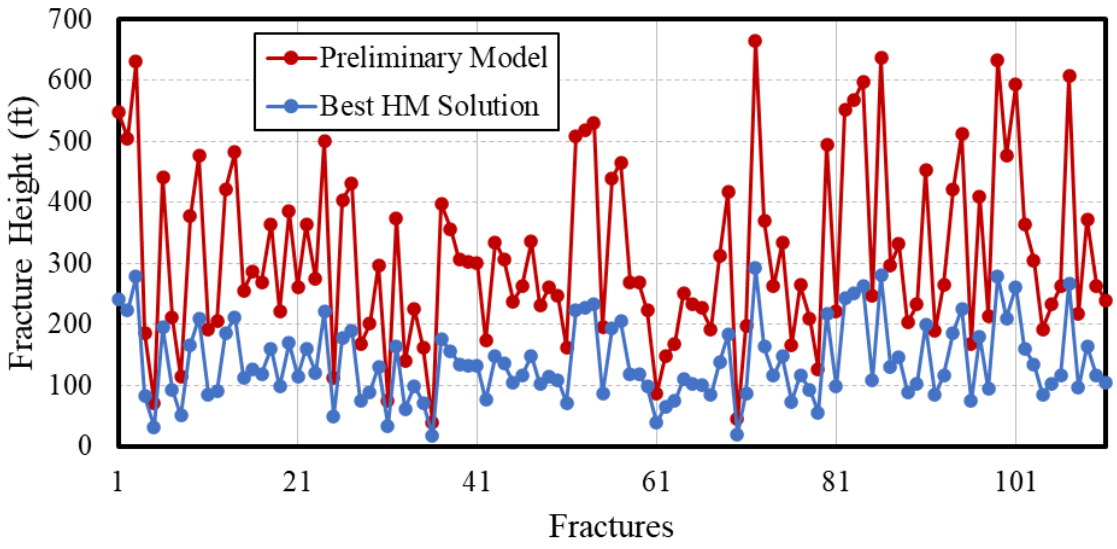

Figure 5.35: Comparing the fracture height in the preliminary fracture model with the fracture height in the calibrated fracture model (best HM solution).

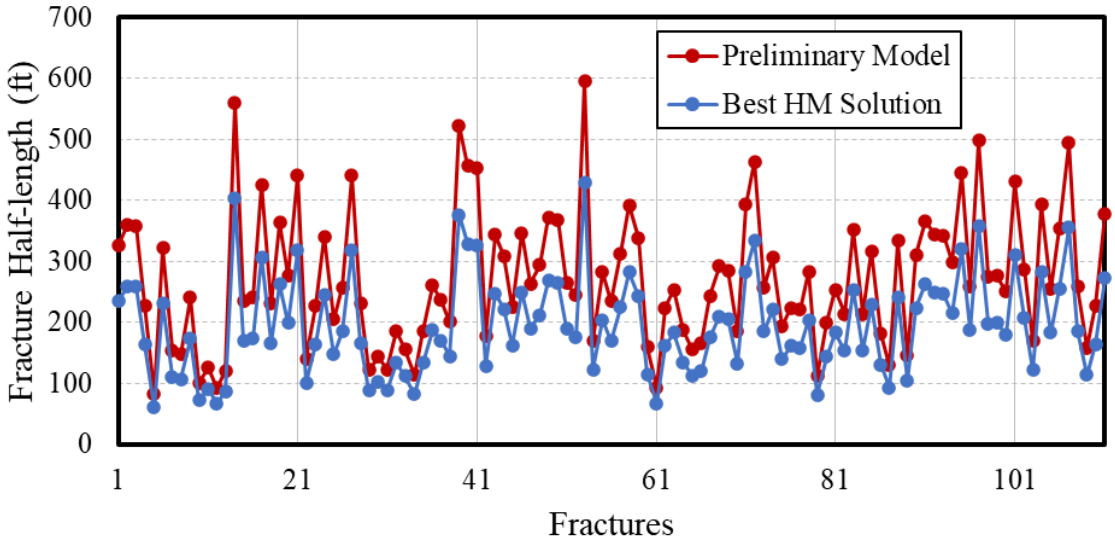

Figure 5.36: Comparing the fracture half-length in the preliminary fracture model with the fracture half-length in the calibrated fracture model (best HM solution).

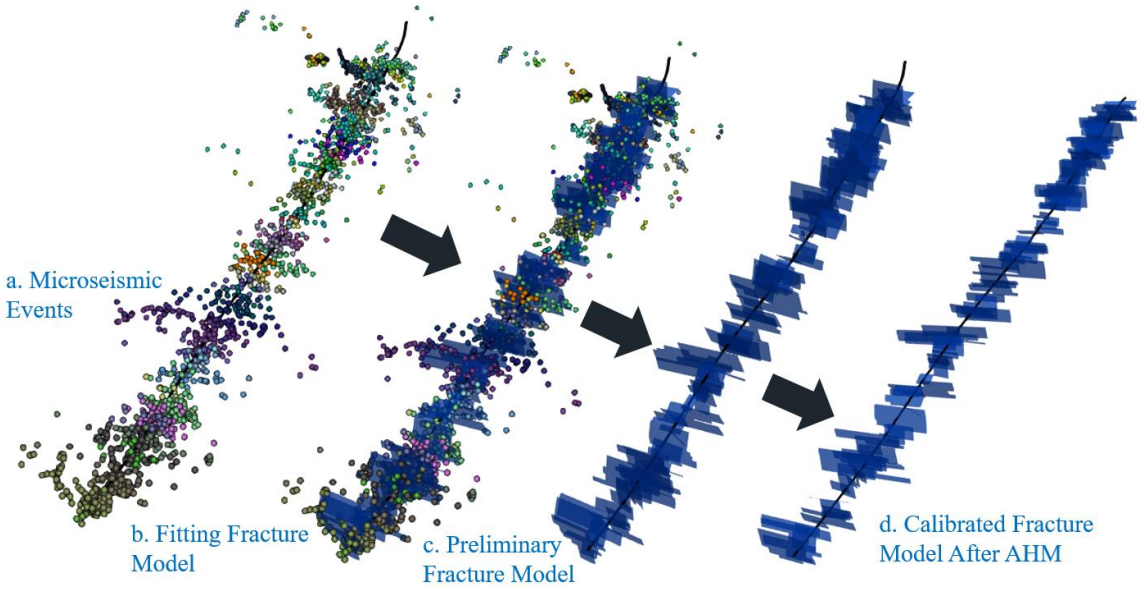

Figure 5.37: Overall process to find the calibrated fracture model. (a) Microseismic events after applying step 1. (b) Microseismic events and the preliminary fracture model are plotted together. (c) Preliminary fracture model (fracture model before applying the scaling factors). (d) Final fracture model after applying the scaling factors from the best HM solution.

Chapter 6: A Case Study in a Shale Gas Reservoir

In this chapter, we present a second case study for the workflow explained in Chapter 3. We obtained a different set of microseismic events from a shale gas reservoir. The stimulated well is a horizontal well that has a 5000 ft horizontal section and 30 stages of hydraulic fractures. In addition, daily production and pressure data are available for the first 76 days.

From a preliminary screening of the microseismic data, **Figure 6.1**, we observed that microseismic events in this case study is more scattered and contains more noise compared to the case study in Chapter 5. Therefore, we decided to simplify our fracture model by fitting only one fracture per stage and selecting rectangular fractures instead of polygonal fractures.

6.1 Stage Definition

The overlapping issue is more severe in this case study and, hence, applying the stage definition step becomes more important, Figure 6.1. As explained in Chapter 3, we use the perforation intervals to plot boundary lines to separate the microseismic events and resolve the overlapping issue, **Figure 6.2**. The modified microseismic events are presented in **Figure 6.3**. We discard the microseismic events that are in a region not assigned to any stage. The length of the perforation interval is not the same for all stages, **Figure 6.4**. Therefore, using the actual perforation intervals is important.

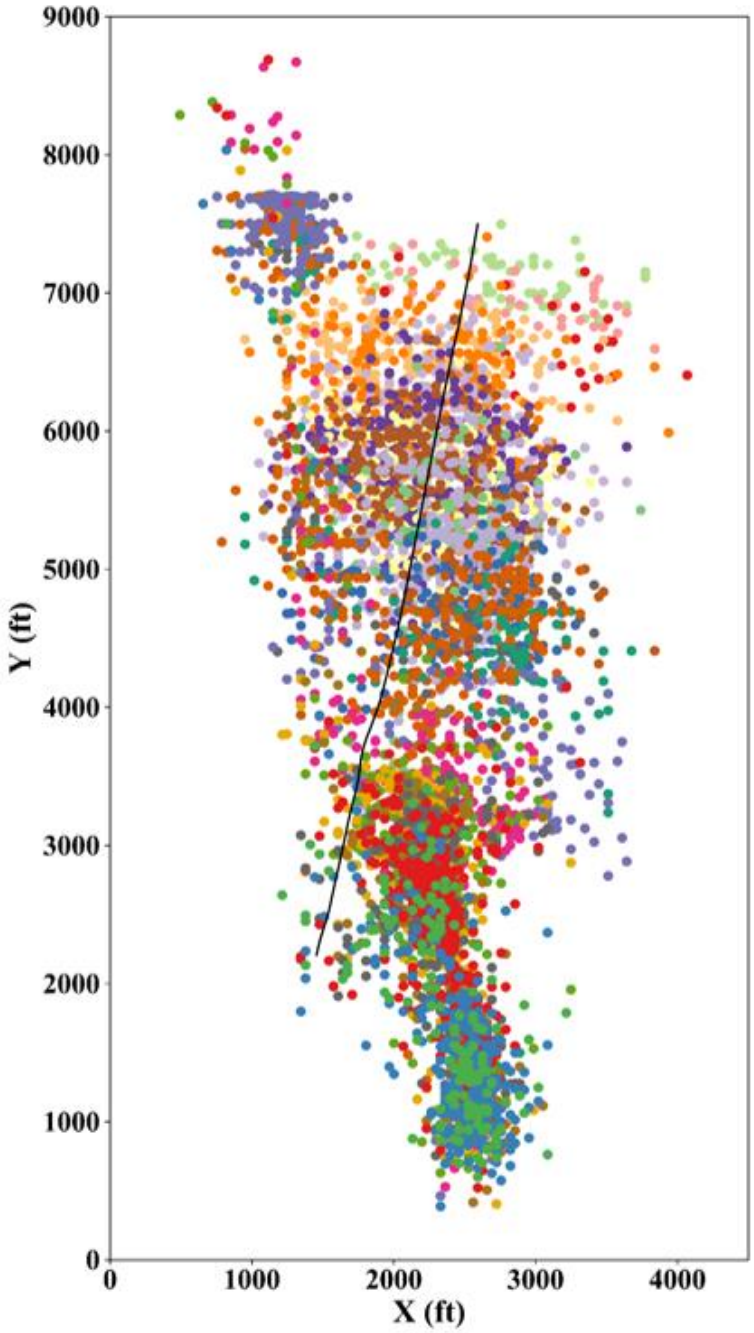

Figure 6.1: Original microseismic events. Microseismic events from different stages are overlapping.

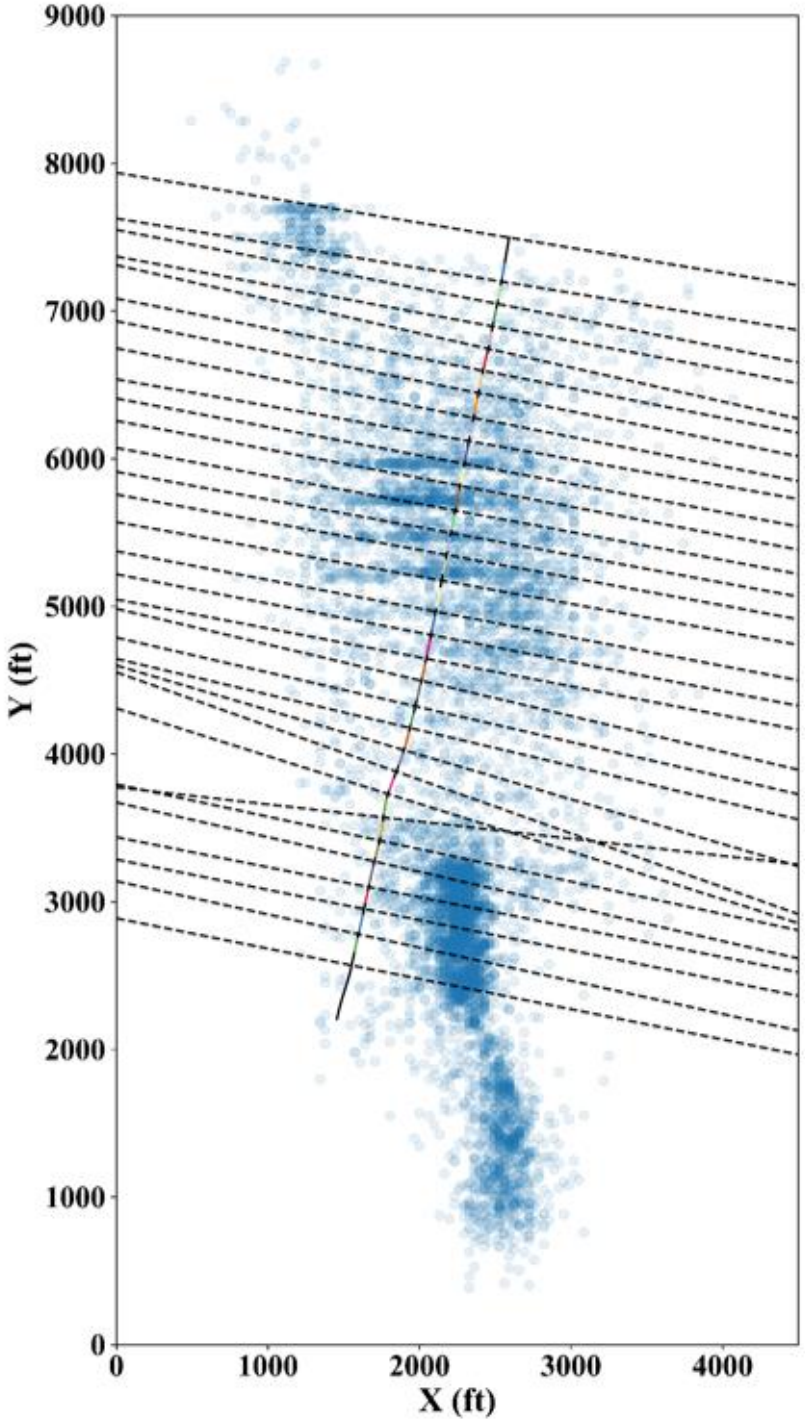

Figure 6.2: Using the perforation intervals to draw boundary lines that are perpendicular to the well path.

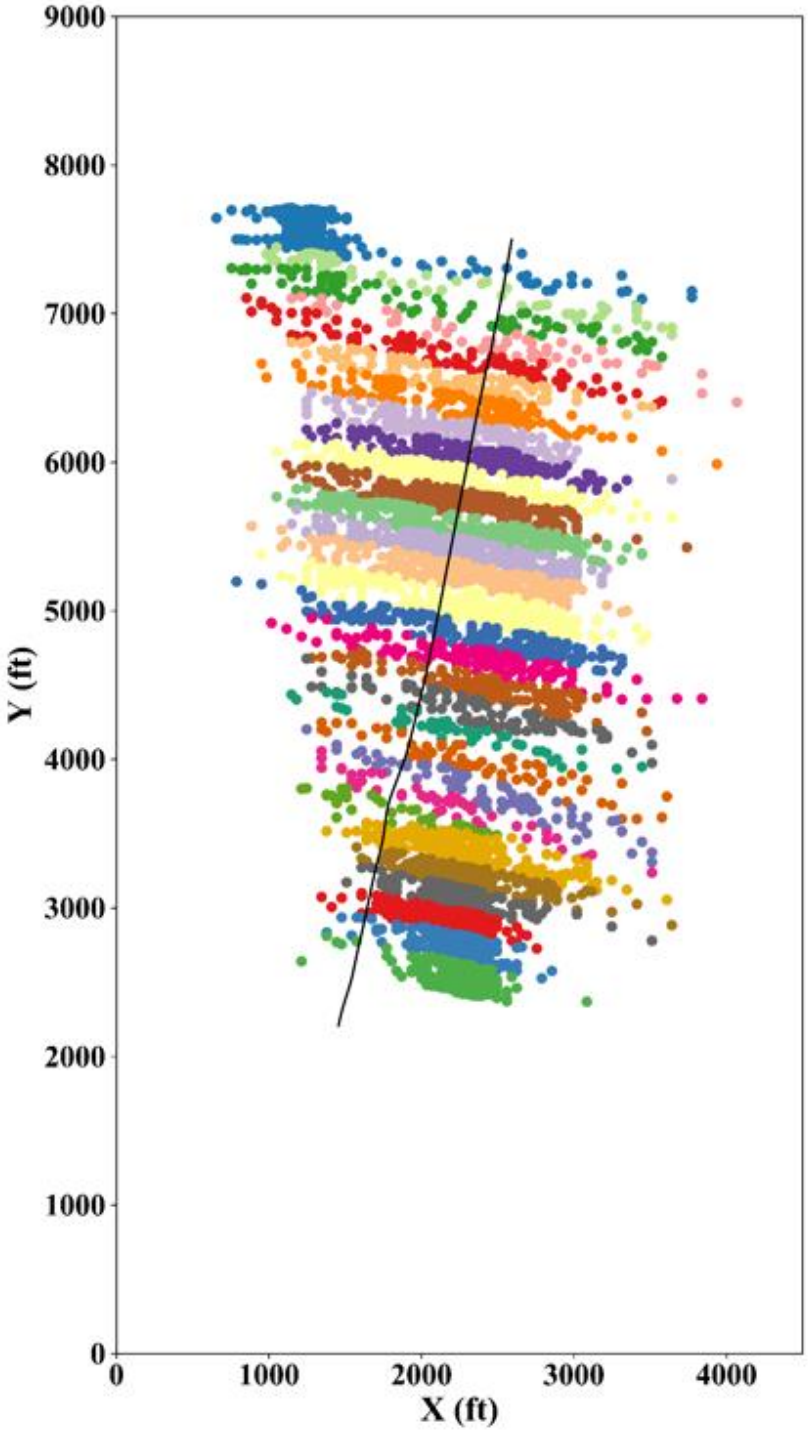

Figure 6.3: Modified microseismic events. Perforation intervals were used to define and separate the stages.

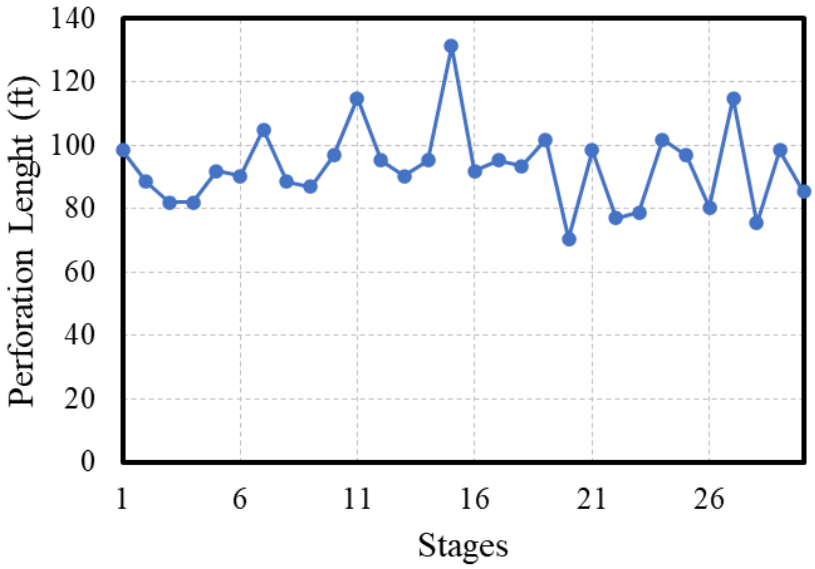

Figure 6.4: Different perforation length for each stage, as indicated from field data.

6.2 CLUSTERING

By conducting the DBSCAN workflow for each stage, some outliers in the microseismic data are eliminated and some clusters were identified. **Figure 6.5** illustrates the number of clusters found by DBSCAN in each stage. DBSCAN requires two input parameters: the radius of investigation (Eps) and the minimum number of points (MinPts). We set Eps to 200 ft and MinPts to 3. Obviously choosing different values of the DBSCAN parameter would affect the fracture model. However, we try to account for this uncertainty in the model calibration step.

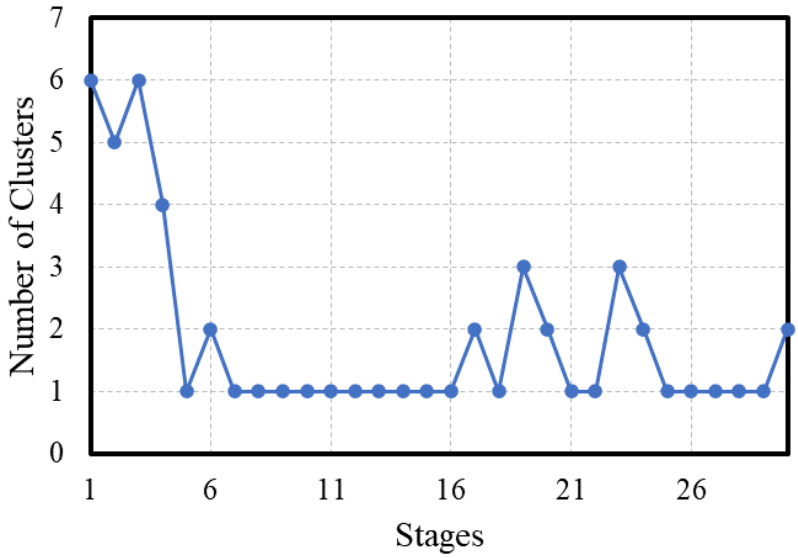

Figure 6.5: Number of clusters found by DBSCAN for each stage. Outliers are not included in any clusters. On average, DBSCAN found 1.9 clusters.

6.3 SURFACE FITTING

We choose one cluster from the previous step and use it to fit a fracture for each stage. We represent the fractures as rectangular planes in the three-dimensional space. Hence, the fractures can have any dip and azimuth angles. We picked three fractures from three random stages to be illustrated in **Figure 6.6**.

The fracture properties such as fracture height, fracture half-length, dip angle, and azimuth angles are shown in **Figure 6.7** to **Figure 6.10**, respectively. The fracture height and half-length are preliminary since they will be adjected in the model calibration step. The average coefficient of determination (R^2) is 0.65 and the R^2 value for each fracture is included in **Figure 6.11**. We summarize the steps to obtain the preliminary fracture model in **Figure 6.12**.

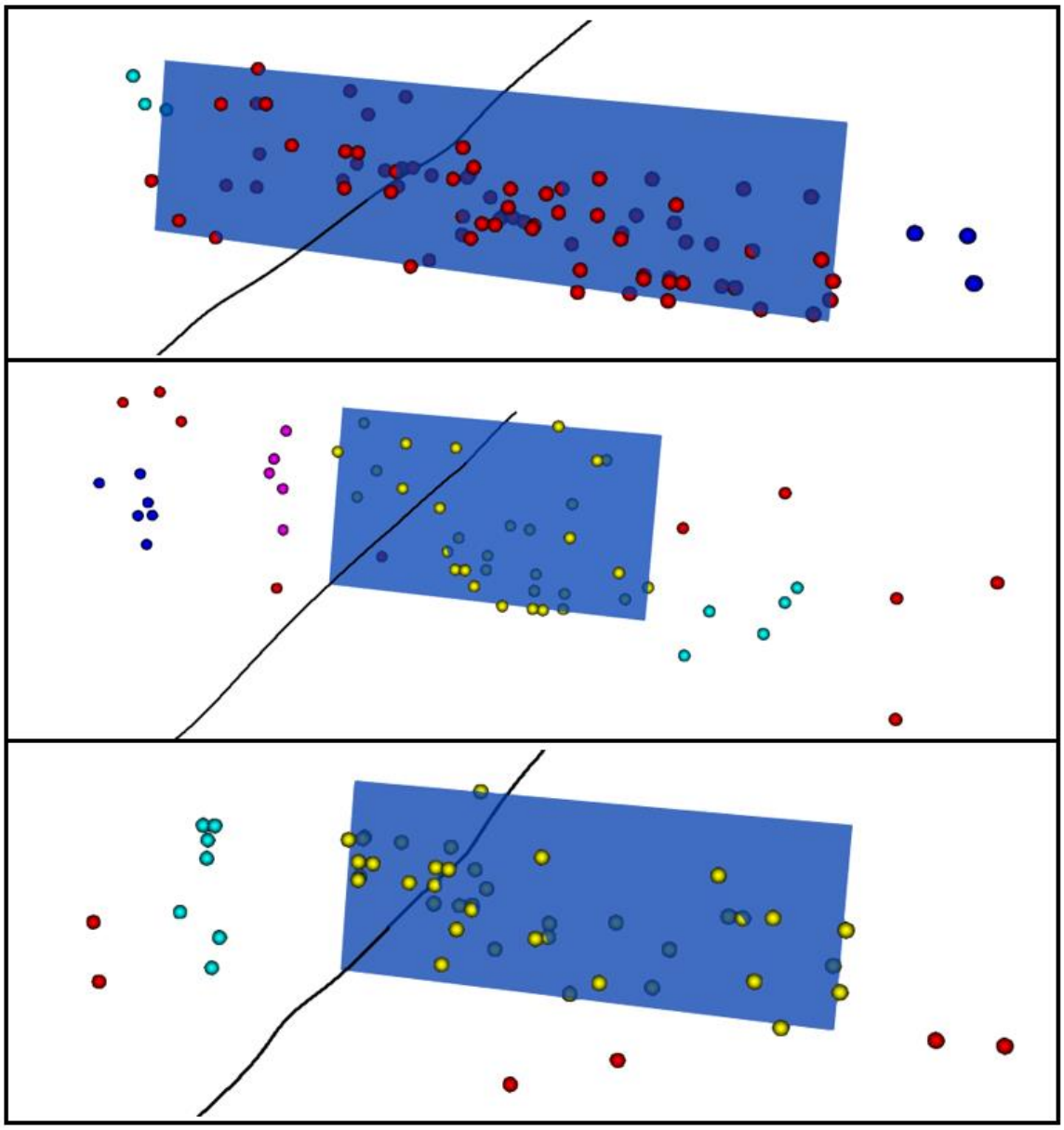

Figure 6.6: Fitting fracture planes in different stages. Three random stages were selected here for illustration.

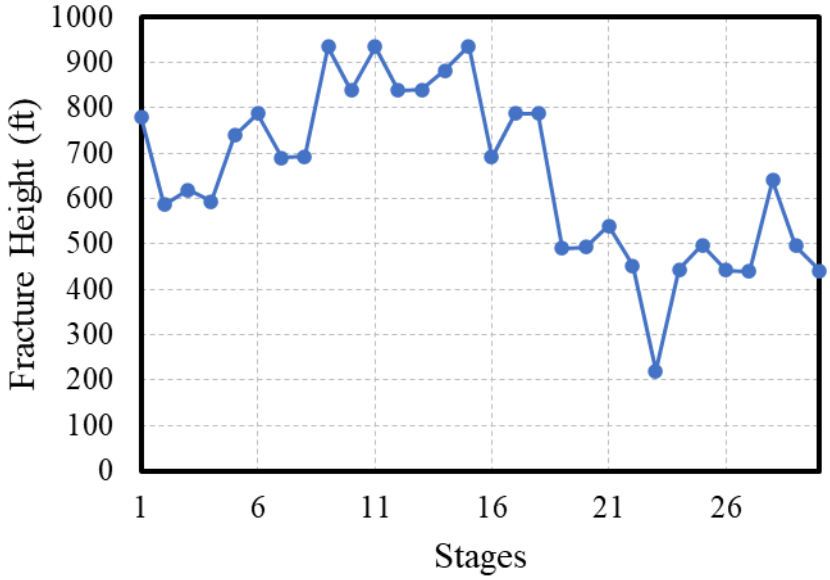

Figure 6.7: Fracture height variations in the preliminary fracture model. Average fracture height is 652 ft.

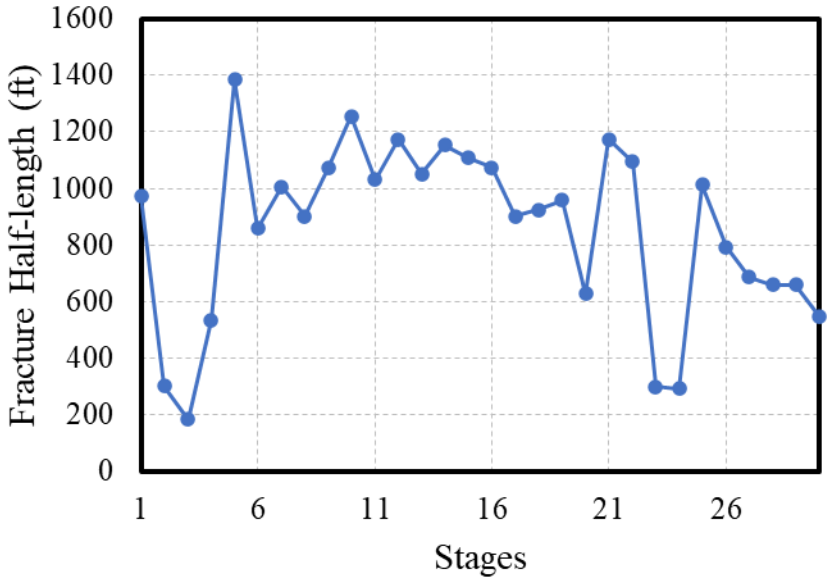

Figure 6.8: Fracture half-length variations in the preliminary fracture model. Average fracture half-length is 856 ft.

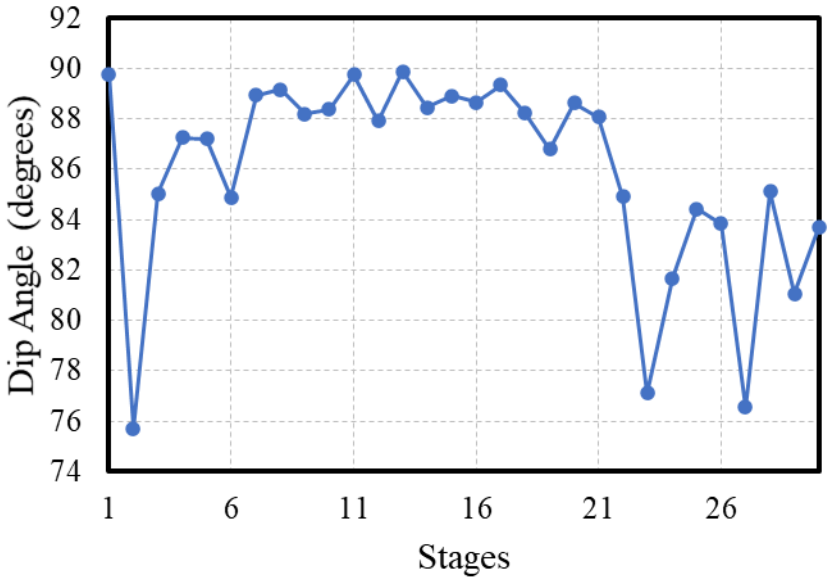

Figure 6.9: Dip angle variations in the fracture model. Average dip angle is 86 degrees.

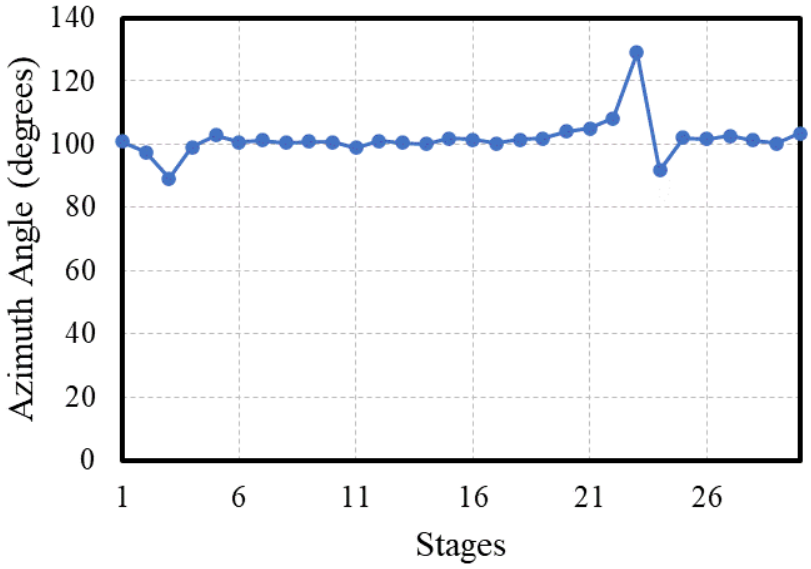

Figure 6.10: Azimuth angle variations in the fracture model. Average azimuth angle is 102 degrees.

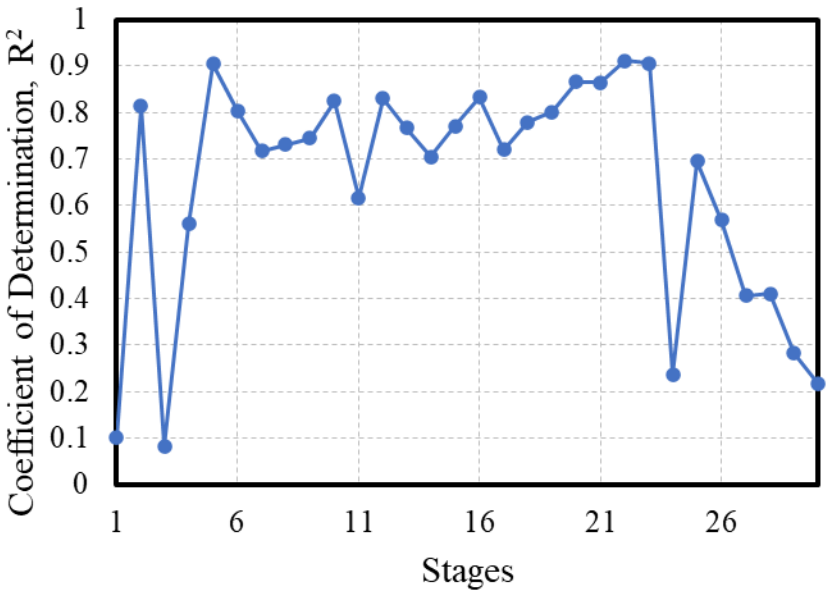

Figure 6.11: Coefficient of determination calculated for each fracture in the fracture model. Average coefficient of determination is 0.65.

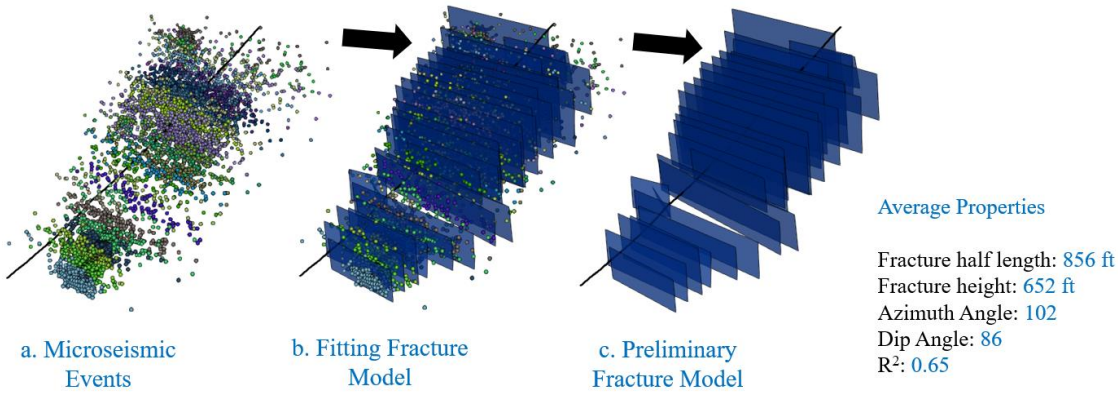

Figure 6.12: Preliminary fracture model. (a) Microseismic events after applying step 1. (b) Microseismic events and the preliminary fracture model are plotted together. (c) Preliminary fracture model (fracture model before applying the scaling factors).

6.4 MODEL CALIBRATION

6.4.1 Reservoir Model

We created a simple reservoir simulation model using the parameters in **Table 6.1**. The reservoir model contains only one well. Building a single-well reservoir model is common in unconventional reservoir modeling if the interference between wells is not strong. Therefore, sector models are more common for single well in tight shale reservoirs. General information about the well and fracture model is available in **Table 6.2**. The reservoir fluid is dry gas and the well was not producing any condensate. Therefore, we assumed that the reservoir fluid is entirely methane.

Table 6.1: Basic parameters used to build the reservoir model.

Parameter	**Value**	**Unit**
Model dimension (x × y × z)	3800 × 5800 × 1400	ft
Number of gird blocks (x × y × z)	38 × 145 × 5	-
Grid block dimension (x × y × z)	100 × 40 × 280	ft
Initial reservoir pressure	8847	psi
Reservoir temperature	215	°F
Porosity	0.056	-
Permeability	50	Nano Darcy
Initial water saturation	0.39	-

Table 6.2: General information about the horizontal well and hydraulic fractures.

Parameter	Value	Unit
Horizontal well length	5000	ft
Number of stages	30	-
Number of fractures	30	-
Fracture spacing	57-381	ft

6.4.2 Assisted History Matching

In the assisted history matching (AHM) workflow, we used five uncertain parameters: the two scaling factors for fracture height and half-length, fracture conductivity, fracture width, and fracture water saturation. **Table 6.3** contains the ranges for each uncertain parameter.

Table 6.3: Uncertain parameters selected to calibrate the preliminary fracture model in the AHM workflow.

Parameter	Min	Max	Unit
Fracture half-length scaling Factor	0.1	1	-
Fracture height scaling factor	0.1	1	-
Fracture conductivity	20	60	md-ft
Fracture width	0.01	5	ft
Fracture water saturation	0.7	0.9	-

The AHM workflow found 18 history matching solutions. The weighted root mean square error (RMSE) for all simulation runs is illustrated in **Figure 6.13**. The solutions below a certain threshold (2000 in this case) are considered history matching solutions.

In **Figure 6.14** and **Figure 6.15**, we compare the production and pressure results of the HM solutions with field data. The well's production was contained in the simulation model by the gas rate from field data. Hence, the simulated gas rate is matching perfectly with field production data. The field bottom hole pressure (BHP) data was obtained from the workflow introduced in Chapter 4. In addition, in **Figure 6.16**, we include the distribution of the uncertain parameters from the HM solutions.

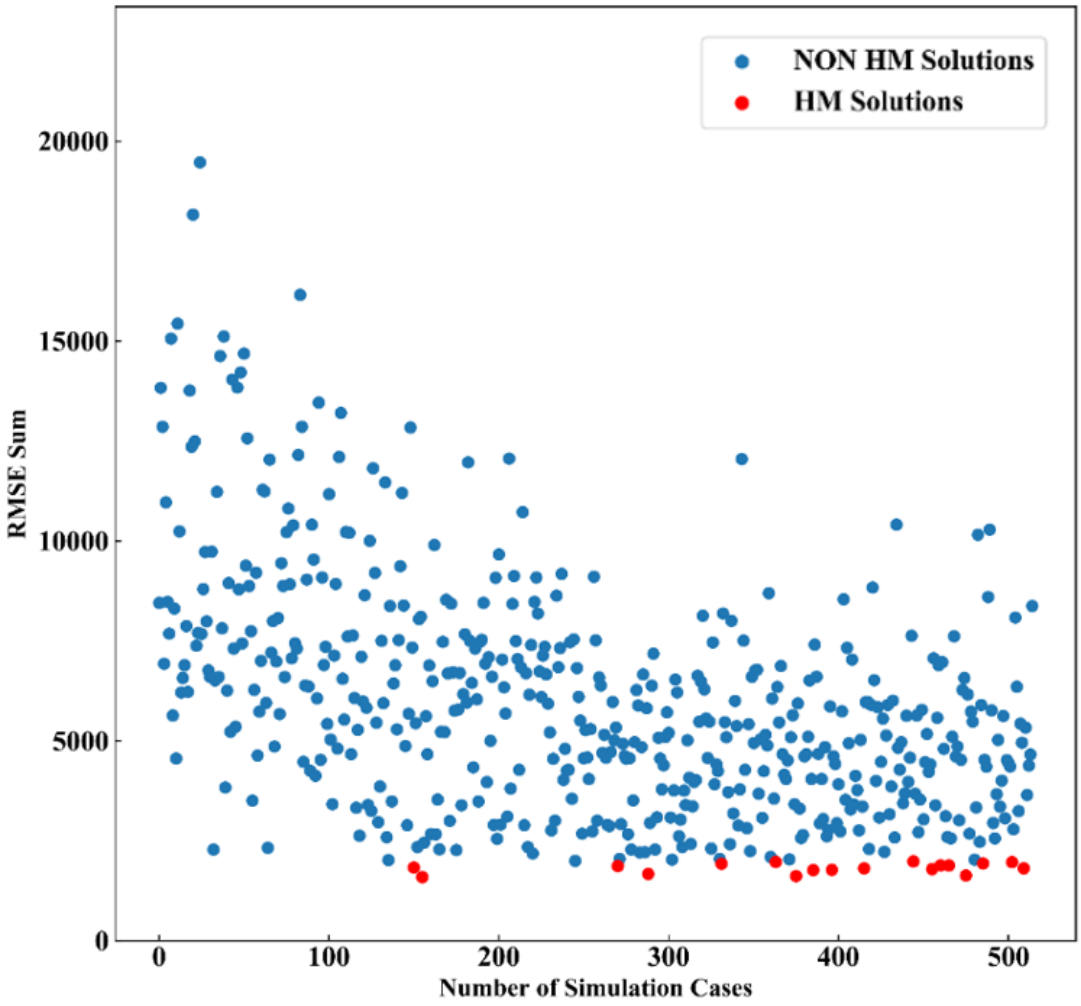

Figure 6.13: Weighted root mean square error (RMSE) for all 515 simulation runs.

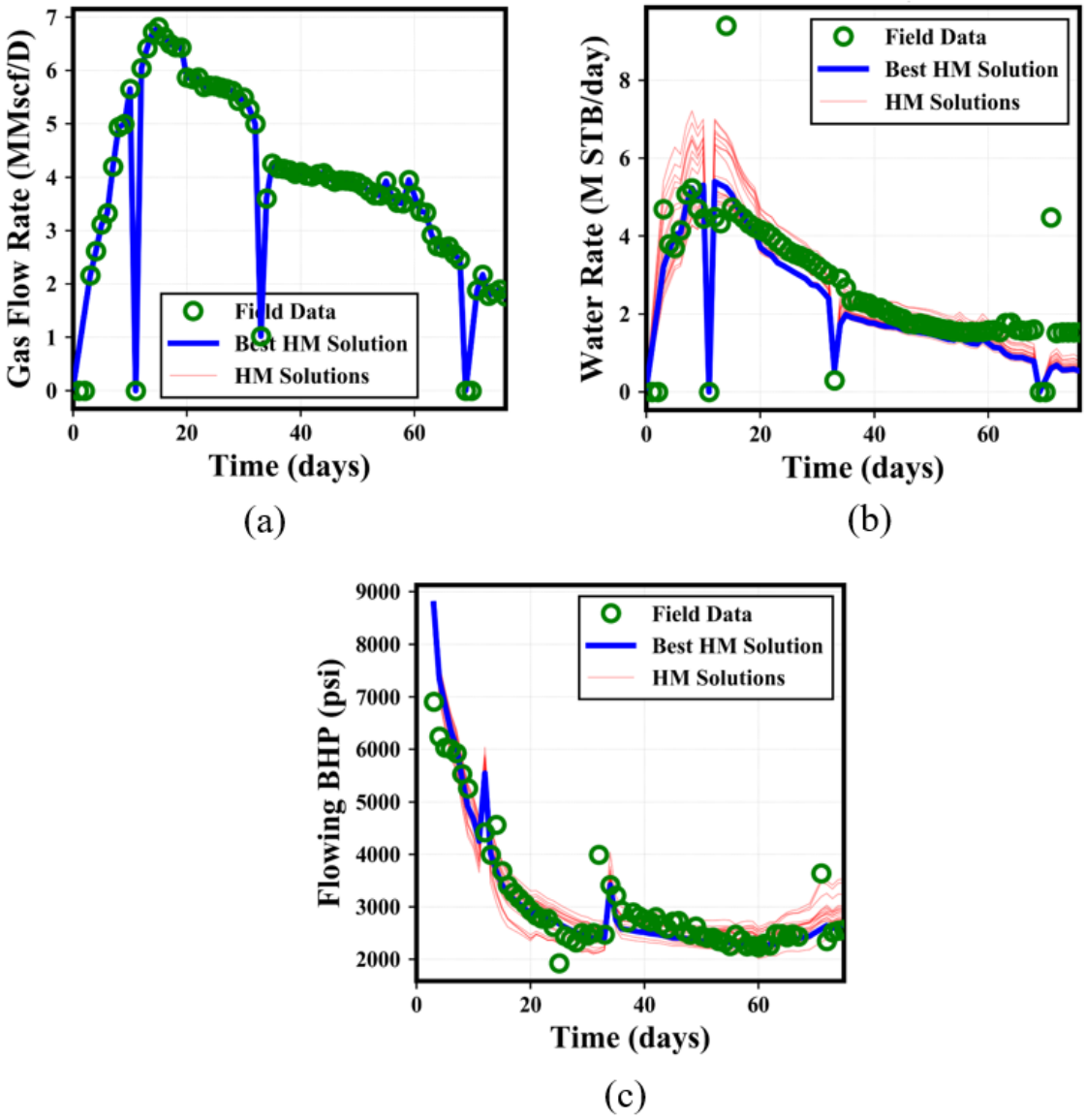

Figure 6.14: Daily production and pressure results from the history matching solutions compared to field data. (a) Gas flow rate. (b) Water flow rate. (c) Flowing bottom hole pressure (BHP).

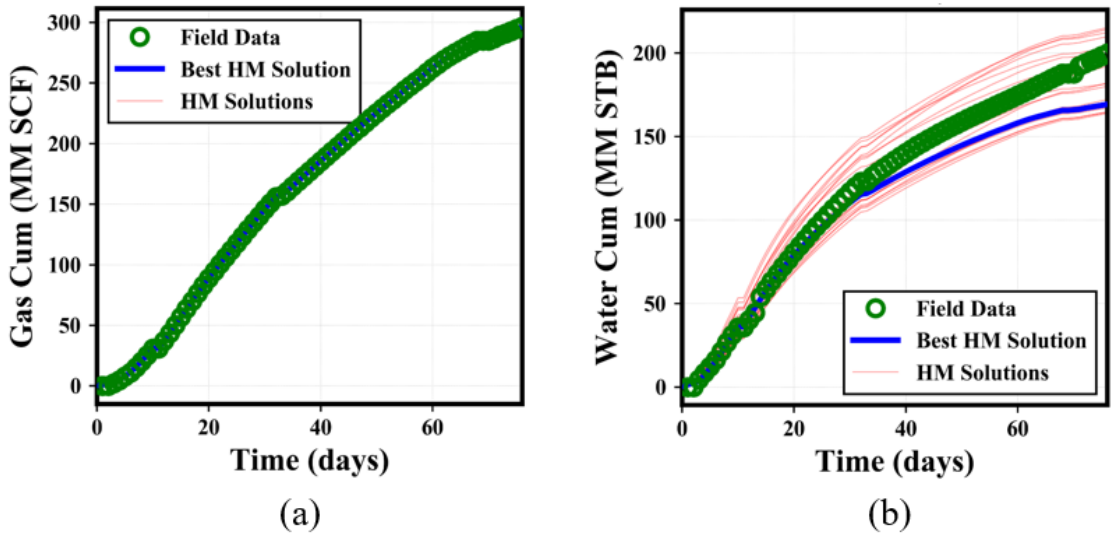

Figure 6.15: Cumulative production data compared to field data. (a) Cumulative gas production. (b) Cumulative water production.

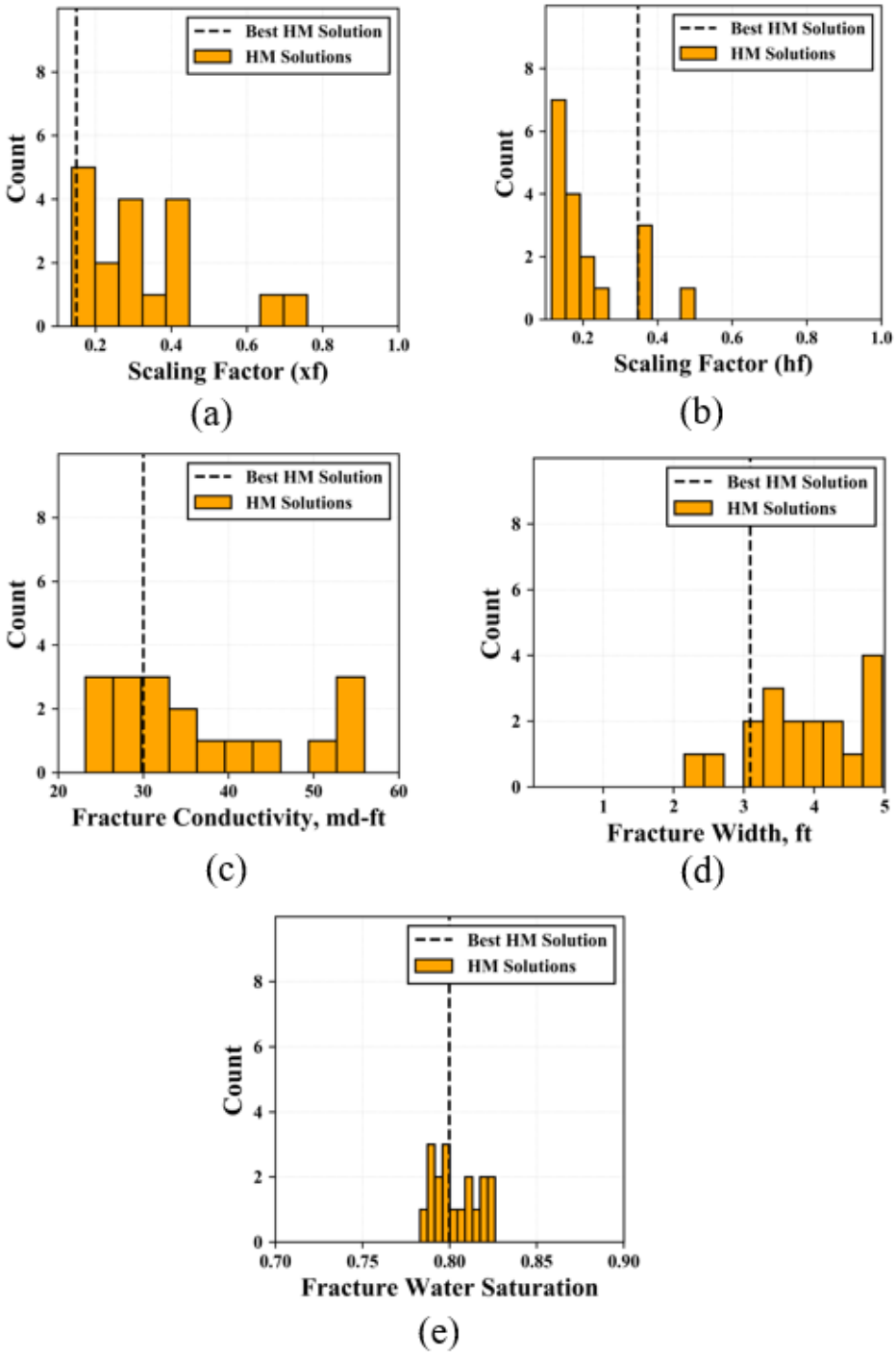

Figure 6.16: Distribution of uncertain parameters in the history matching solutions. (a) Scaling factor for fracture half-length. (b) Scaling factor for fracture height. (c) Fracture conductivity. (d) Fracture width. (e) Fracture water saturation.

The final step is to modify the fracture model using the scaling factors in the HM solutions. We chose the scaling factors from the best HM solution to adjust our fracture model. For the best HM solution, the scaling factors are 0.35 and 0.15 for fracture height and half-length, respectively. In **Figure 6.17** and **Figure 6.18**, we compare the fracture height and half-length, respectively, in the preliminary fracture model with the final calibrated fracture model. The preliminary fracture model was overestimating the fracture height and half-length. The field production data helped us identify the appropriate fracture height and half-length.

The entire process of build a fracture model from microseismic events is summarized in **Figure 6.19**. A preliminary fracture model was first fitted to the microseismic events, Figure 6.19 (b). After identifying the appropriate scaling factor from the AHM workflow, the fracture model is modified by reducing the fracture height and half-length, Figure 6.19 (d).

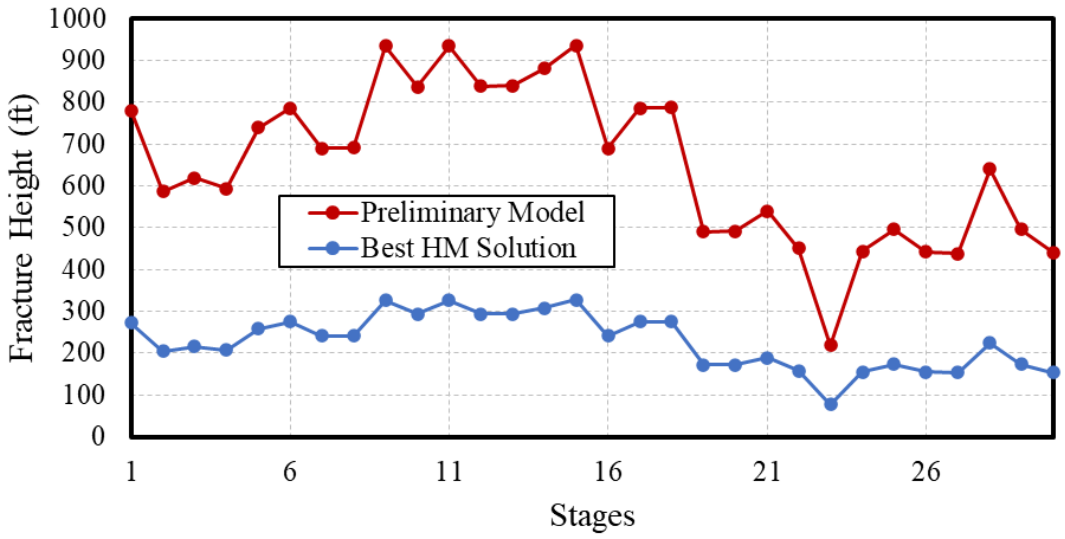

Figure 6.17: Comparing the fracture height in the preliminary fracture model with the fracture height in the calibrated fracture model (best HM solution).

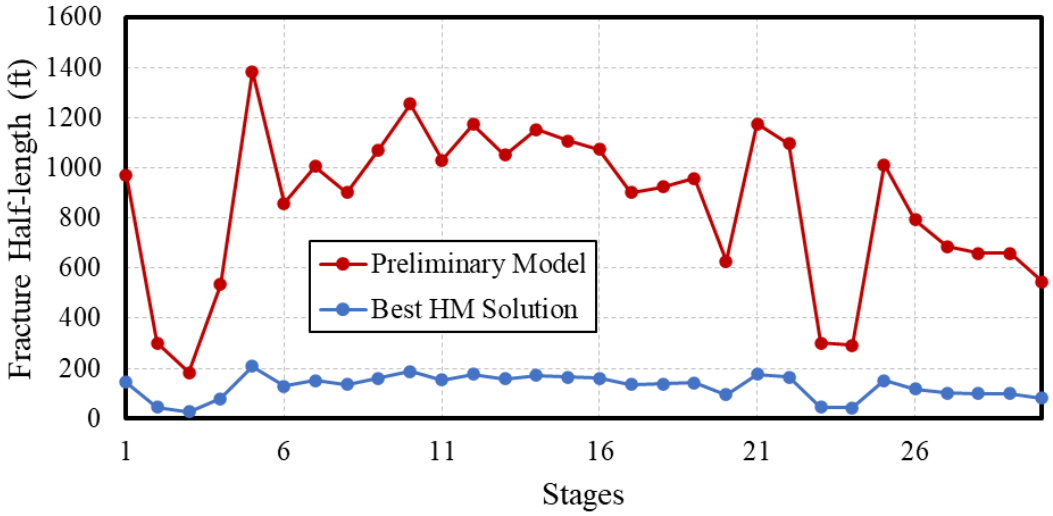

Figure 6.18: Comparing the fracture half-length in the preliminary fracture model with the fracture half-length in the calibrated fracture model (best HM solution).

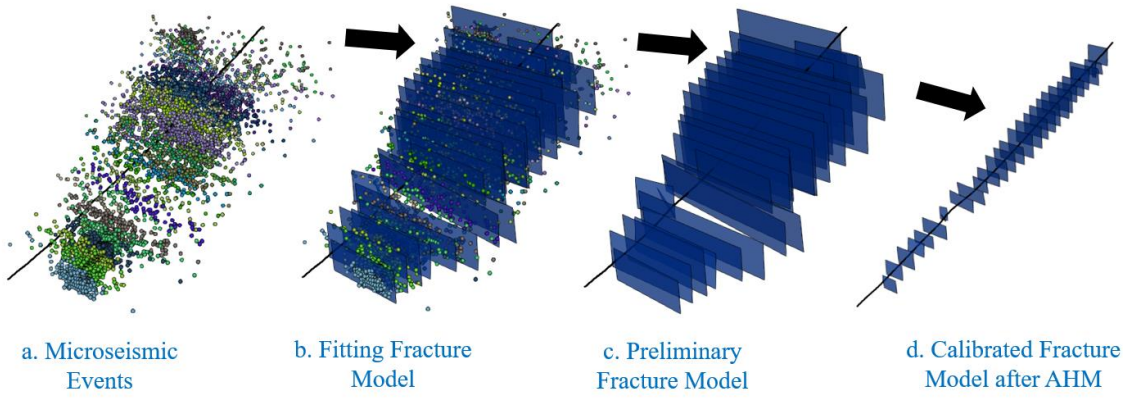

Figure 6.19: Overall process to find the calibrated fracture model. (a) Microseismic events after applying step 1. (b) Microseismic events and the preliminary fracture model are plotted together. (c) Preliminary fracture model (fracture model before applying the scaling factors). (d) Final fracture model after applying the scaling factors from the best HM solution.

Chapter 7: Summary, Conclusions, and Recommendations for Future Work

7.1 Fracture Modeling Using Microseismic Events

7.1.1 Summary and Conclusions

In this thesis, we introduced a direct method to build a discrete fracture model using microseismic events. The method is direct because it relies on measured data such as microseismic events, wellbore trajectory, perforation intervals, and field production data. The method consists of five steps: stage definition, clustering, surface fitting, embedding the fracture model into a reservoir model, and model calibration.

We first separate the microseismic events using the perforation intervals. We draw boundary lines that are perpendicular to the well path to define the region for each stage. Then, for each stage, we perform a density-based cluster method known as DBSCAN. This step helps us eliminate outliers in the microseismic events and identify some clusters in each stage.

Next, we choose one cluster from the previous step and fit a fracture plane to the microseismic events. The fracture plane should minimize the square error function. If we decide to fit more than one fracture per stage, we use DBSCAN again to find clusters based on the error distribution from surface fitting. Then, we fit new a fracture plane for each cluster. In addition, our workflow can define the boundary of the fracture plane as polygons based on the distribution of the microseismic events. After that, we insert the fracture model into a reservoir simulation mode using embedded discrete fracture model (EDFM) method. EDFM is an efficient method to represent the effect of fractures in reservoir simulation models.

The last step of the workflow is to calibrate the fracture model using two scaling factors. The first scaling factor reduces the fracture height and the second reduces the fracture half-length. We rely on an assisted history matching (AHM) workflow to determine the appropriate values for each scaling factor. We include the two scaling factors as uncertain parameters in the AHM workflow. The AHM workflow is an iterative process that uses a proxy model to design the simulation runs for the next iteration. The AHM workflow will determine the range for each scaling factor that can be used to match the field production data.

Results from Chapters 5 and 6 indicate that the preliminary fracture model (the fracture model before calibration) overestimates the size of the fractures. Hence, the calibration step is important to generate realistic fracture models. In addition, using more fractures per stage leads to smaller fractures. In Chapter 5, we evaluated two scenarios: one with a single fracture per stage and the other with multiple fractures per stage. Adding more fractures per stage resulted in smaller scaling factors. More fractures add to the total fracture surface area and, hence, smaller fractures are needed to match the same production data.

7.1.2 Recommendations for Future Work

The method introduced in this thesis can be improved by implementing the followings:

- Even though we included a step to fit multiple fractures per stage, the method is unable to fit many fractures per stage. For example, the average cluster efficiency from Chapter 5 was only 14%. We suggest modifying the step that identifies clusters in each stage.

- We assumed that all the microseismic events around the wellbore are caused by hydraulic fractures. However, some microseismic events are generated by activation of natural fractures. We recommend finding a method to separate the two types before fitting the fractures.
- We recommend performing sensitivity study to evaluate the effect of adding natural fracturs around the wellbore.
- Some microseismic events are induced in a shallower or deeper formation and not in the targeted reservoirs. However, our current workflow does not account for reservoir boundaries. Therefore, we recommend integrating reservoir boundaries in the workflow and using only the microseismic events that are induced in the targeted reservoir.

7.2 Converting Wellhead Pressure to Bottom Hole Pressure

7.2.1 Summary

In Chapter 4, we present a novel approach to convert static wellhead pressure to bottom hole pressure in wells that produce gas and water. The main assumption in our approach is that both gas and water will not flow back to the formation after shutting the well from surface.

We consider a wellbore that has a gas column overlaying a water column. The gas column is discretized into (n) segments and the pressure at each segment is treated as unknown. The depth of the gas-water interface is also added to the unknowns. Then, the unknowns are found by solving (n+1) nonlinear equations using Newton-Raphson method.

7.2.2 Recommendations for Future Work

The method can be further improved by applying the followings:

- We considered only static wellhead pressure. However, companies are also interested in converting flowing wellhead pressure to flowing bottom hole pressure. We recommend expanding the method to include flowing wellhead pressure as well.
- In some gas wells, condensate comes out of solution in the wellbore because of the lower pressure in the wellbore compared to the reservoir. Therefore, we recommend considering condensate in the wellbore to generalize the method.

Appendix A

We include here the derivation for Equations 4 and 5 from Chapter 3. Given the equation of a plane, **Equation 40**, we would like to calculate the dip and azimuth angles.

$$y = a + bx + cz \tag{40}$$

DIP ANGLE

First, we find the equation of the normal vector for the plane. We move all the terms in Equation 40 to one side. The coefficients will be the same for the normal vector, **Equation 41**.

$$n = b\hat{\imath} - \hat{\jmath} + c\hat{k} \tag{41}$$

From the schematic in **Figure A.1**, the dip angle is the same as the angle between the normal vector and the positive or negative $\hat{k}$ unit vector. Therefore, we can calculate the dip angle (θ) by calculating the angle between the normal vector and the $\hat{k}$ unit vector, Equation 42. Then, by substituting for the dot product and the magnitudes for the vectors, we obtain **Equation 43**.

$$\theta = cos^{-1}\left[\frac{n \cdot \hat{k}}{\|n\|\|\hat{k}\|}\right] \tag{42}$$

$$\begin{aligned} \theta &= cos^{-1}\left[\frac{c}{\sqrt{b^2 + c^2 + 1}}\right] && for\ c > 0 \\ \theta &= cos^{-1}\left[\frac{-c}{\sqrt{b^2 + c^2 + 1}}\right] && for\ c < 0 \end{aligned} \tag{43}$$

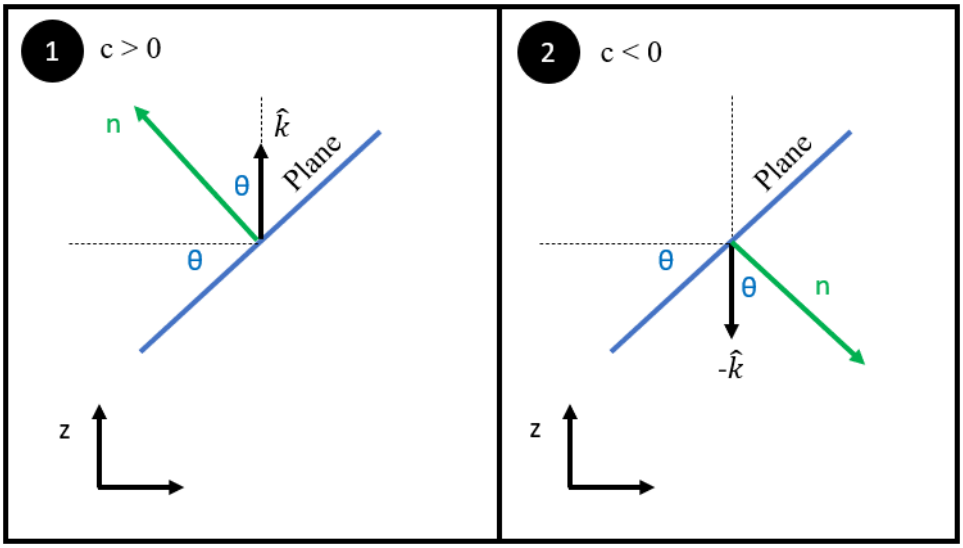

Figure A.1: Schematic for calculating the dip angle for two cases. (1) When the coefficient "c" positive. (2) When the coefficient "c" is negative.

AZIMUTH ANGLE

To calculate the azimuth angle (α), we use the equation of the normal vector again. However, this time, we consider the vector in the x-y plane, **Equation 44**.

$$n = b\hat{\imath} - \hat{\jmath} \tag{44}$$

We define the azimuth angle as the angle from the north or the positive y-axis. **Figure A.2** shows the schematic for the two possible cases in the x-y plane. The coefficient (b) is positive in the first case and negative in the second case. Using the definition of the tangent function, **Equation 45**, and by substituting for the x and y component from the normal vector, we derive an equation for the azimuth angle, **Equation 46**.

$$\begin{aligned} \alpha &= tan^{-1}\left[\frac{n_y}{n_x}\right] && for\ b > 0 \\ 180 - \alpha &= tan^{-1}\left[\frac{n_y}{n_x}\right] && for\ b < 0 \end{aligned} \tag{45}$$

$$\alpha = tan^{-1}\left[\frac{-1}{b}\right] \qquad for\ b > 0$$
$$\alpha = 180 - tan^{-1}\left[\frac{-1}{b}\right] \qquad for\ b < 0 \tag{46}$$

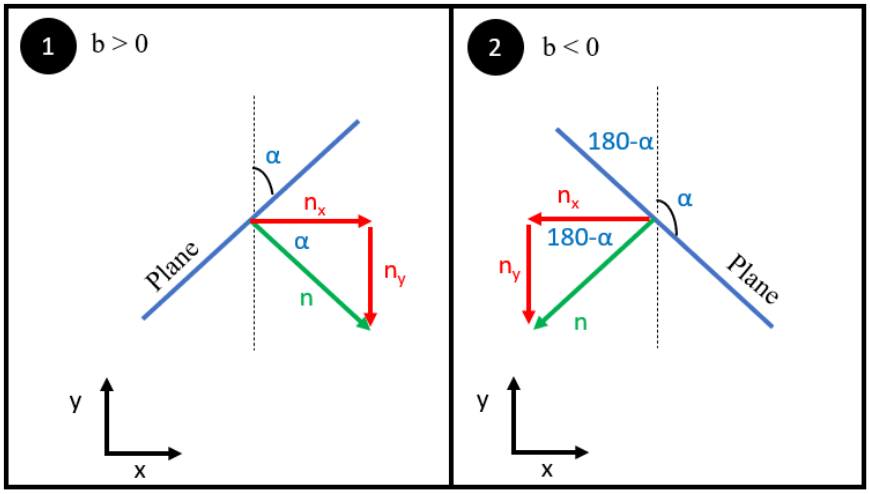

Figure A.2: Schematic for calculating the azimuth angle for two cases. (1) When the coefficient "b" is positive. (2) When the coefficient "b" is negative.

Appendix B

In this appendix, we illustrate how our cutting functions work. we represent the fractures as polygons in the three-dimensional space. We store the coordinates of the vertices for each polygon. Therefore, our cutting function should modify the location of some of the vertices and keep other vertices the same.

CUTTING IN THE FRACTURE HEIGHT DIRECTION

We use one vector as a reference to cut the fracture in the fracture height direction, **Figure B.**. The reference vector is easy to extract from the polygon data because it consists of the first two points in any polygon. All our polygons start at the shallowest point and end at the same depth. **Equations 47-49** represent a vector that has the same direction as our reference vector, but different starting and ending points. The starting point is any vertex in the polygon (x_j, y_j, z_j) and the end point will be the new location of that vertex. The end point can be calculated by setting the z component, Equation 49, to the new depth (z_{new}) that will achieve the targeted fracture height or by solving **Equation 50** for the parameter (t).

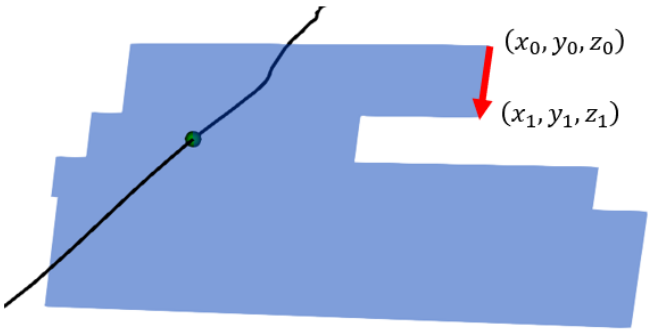

Figure B.1: Vector used to cut a fracture polygon in the fracture height direction.

$$x = x_j + t\,(x_1 - x_0) \tag{47}$$

$$y = y_j + t\,(y_1 - y_0) \tag{48}$$

$$z = z_j + t\,(z_1 - z_0) = z_{new} \tag{49}$$

$$t = \frac{z_{new} - z_j}{z_1 - z_0} \tag{50}$$

CUTTING IN THE FRACTURE HALF-LENGTH DIRECTION

To cut the fracture in the fracture half-length direction, we rely on the mathematical principle that the dot product of any two orthogonal vectors is zero. There are two main steps we follow.

Step 1: Finding the Point C

From **Figure B. (a)**, the vector (r_1) is known because it consists of the deepest two points in the polygon, **Equation 52**. The point C lies on the vector (r_1); therefore, we can write its coordinates using the equation of the vector (r_1) and the unknown parameter (k), **Equation 53**. We can also write an equation of the second vector (r_2) in terms of the unknown parameter (k), **Equation 54**. The vectors (r_1 and r_2) are orthogonal, hence, their dot product is equal to zero, **Equation 51**. By applying the dot product and setting it to zero, we can solve for the parameter (k), **Equation 55**. After that, we can substitute the parameter (k) back into Equation 53 to find the point C.

Step 2: Finding the Points P

From **Figure B. (b)**, the vector (r_2) is now known from the previous step, **Equation 57**. The point P lies on the vector (r_2); therefore, we can write its coordinates using the equation of the vector (r_2) and the unknown parameter (m), **Equation 58**. The point P will be different for each vertex in the polygon. We can also write an equation of the second

vector (r_3) in terms of the unknown parameter (m), **Equation 59**. The vectors (r_2 and r_3) are orthogonal, hence, their dot product is equal to zero, **Equation 56**. By applying the dot product and setting it to zero, we can solve for the parameter (m), **Equation 60**. After that, we can substitute the parameter (m) back into Equation 58 to find the point P.

Once we have the point P for each vertex, we calculate the distance between the vertex and its point P. If the distance is higher than the targeted distance, we move the vertex along the vector (r_3) until the targeted distance is achieved.

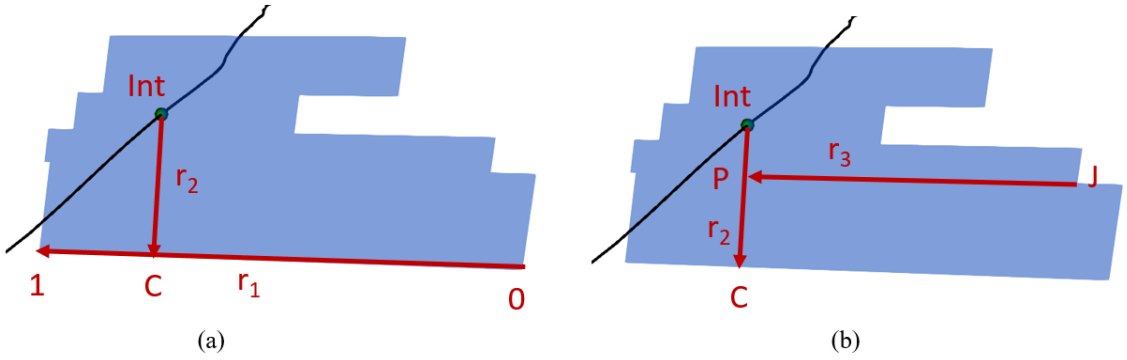

Figure B.2: Schematic for cutting the polygon in the fracture half-length direction.

Equations for the First Step (Finding the Point C)

$$r_1 \cdot r_2 = 0 \tag{51}$$

$$r_1 = \langle x_0, y_0, z_0 \rangle + t\langle \Delta x,\ \Delta y,\ \Delta z \rangle \tag{52}$$

$$C(x_0 + k\Delta x,\ y_0 + k\Delta y,\ z_0 + k\Delta z) \tag{53}$$

$$r_2 = \begin{vmatrix} x_0 + k\Delta x - x_{int}, \\ y_0 + k\Delta y - y_{int}, \\ z_0 + k\Delta z - z_{int} \end{vmatrix} \tag{54}$$

$$k = \frac{(x_{int} - x_0)\Delta x + (y_{int} - y_0)\Delta y + (z_{int} - z_0)\Delta z}{\Delta x^2 + \Delta y^2 + \Delta z^2} \tag{55}$$

Equations for the Second Step (Finding the Point P)

$$r_2 \cdot r_3 = 0 \tag{56}$$

$$r_2 = \langle x_{int}, y_{int}, z_{int} \rangle + t\langle \Delta x,\ \Delta y,\ \Delta z \rangle \tag{57}$$

$$P(x_{int} + m\Delta x,\ y_{int} + m\Delta y,\ z_{int} + m\Delta z) \tag{58}$$

$$r_3 = \left\langle \begin{matrix} x_{int} + m\Delta x - x_j, \\ y_{int} + m\Delta y - y_j, \\ z_{int} + m\Delta z - z_j \end{matrix} \right\rangle \tag{59}$$

$$m = \frac{(x_j - x_{int})\Delta x + (y_j - y_{int})\Delta y + (z_j - z_{int})\Delta z}{\Delta x^2 + \Delta y^2 + \Delta z^2} \tag{60}$$

Appendix C

Here we derive an equation to convert pressure from one point to another in a gas column. This derivation is also available in Cullender and Smith (1956). We start with the conservation of energy law, **Equation 61**. We assume there is no energy loss and zero kinetic energy (static condition). Next, we consider the ideal gas law with a z-factor correction, **Equation 62**. We can also write the gas law in terms of mass and molar mass, **Equation 63**.

$$V\,dp\; - m\,g\,dD = 0 \tag{61}$$

$$V = \frac{znRT}{p} \tag{62}$$

$$V = \frac{zmRT}{M\,p} \tag{63}$$

By combining Equations 61 and 63, we obtain **Equation 64**. The mass variable can be canceled because it appears in both terms in Equation 64. Then, we integrate the equation from the wellhead conditions to the bottom hole condition, **Equation 65**. The left-hand side of the equation can be integrated easily, **Equation 66**. After that, we re-write the equation in terms of the specific gravity, **Equations 67** and **68**. Finally, we convert the equation to be used in field units, **Equation 69**.

$$\frac{m\,R\,T\,z}{M\,p}\,dp\; - m\,g\,dD = 0 \tag{64}$$

$$\int_0^D g\,dD = \int_{p_w}^{p_s} \frac{R\,T\,z}{M\,p}\,dp \tag{65}$$

$$\frac{M}{R}\,g\,D = \int_{p_w}^{p_s} \frac{T\,z}{p}\,dp \tag{66}$$

$$\gamma_g = \frac{M}{M_{air}} \tag{67}$$

$$\frac{M_{air}\, g}{R}\, \gamma_g\, D = \int_{p_w}^{p_s} \frac{T\, z}{p}\, dp \tag{68}$$

$$\frac{\gamma_g\, D}{53.351} = \int_{p_w}^{p_s} \frac{T\, z}{p}\, dp \tag{69}$$

Glossary

NOMENCLATURE (CHAPTER 3)

m	=	Number of microseismic events in one cluster
x_k, y_k, z_k	=	x, y, and z coordinates for the microseismic events, ft
$E(g)$	=	Objective or error function
$g(x, y)$	=	Guess or base function
θ	=	Dip angle, degrees
α	=	Azimuth angle, degrees

NOMENCLATURE (CHAPTER 4)

r	=	Wellbore radius, ft
V_{std}	=	Gas volume at standard conditions
V_w	=	Water volume at standard conditions
T_{std}	=	Standard temperature
p_{std}	=	Standard pressure
z	=	Gas compressibility factor or z-factor

NOMENCLATURE (APPENDICES)

V	=	Gas volume
p	=	pressure
m	=	mass
g	=	Gravitational constant
D	=	Depth
R	=	Universal gas constant

T = Temperature

M = Molecular weight

n = Number of moles

References

Aziz, K. 1967. Calculation of Bottom-Hole Pressure in Gas Wells. *J Pet Technol* **19** (7): 897–899. SPE-1676-PA. https://doi.org/10.2118/1676-PA.

Cullender, M.H. and Smith, R.V. 1956. Practical Solution of Gas-Flow Equations for Wells and Pipelines with Large Temperature Gradients. *Transactions of the AIME*, 207 (01). SPE-696-G. doi.org/10.2118/696-G.

Dawkins, P. 2020. Approximating Definite Integrals. https://tutorial.math.lamar.edu/classes/calcii/approximatingdefintegrals.aspx (accessed 26 July 2020).

Dean, R. and Lo, L. 1988. Simulations of Naturally Fractured Reservoirs. *SPE Res Eng* **3** (2): 638–648. SPE-14110-PA. https://doi.org/10.2118/14110-PA.

Dranchuk, P. and Abou-Kassem, H. 1975. Calculation of Z Factors for Natural Gases Using Equations of State. *J Can Pet Technol*. **14** (3): PETSOC-75-03-03. https://doi.org/10.2118/75-03-03.

Ester, M., Kriegel, H., Sander, J. et al. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Database with Noise. Paper presented at the Second International Conference on Knowledge Discovery and Data Mining, Portland, Oregon, 2—4 August.

Karimi-Fard, M., Durlofsky, L. and Aziz, K. 2004. An Efficient Discrete-Fracture Model Applicable for General-Purpose Reservoir Simulators. *SPE J.* **9** (2): 227–236. SPE-88812-PA. https://doi.org/10.2118/88812-PA.

Li, L. and Lee, S. H. 2008. Efficient Field-Scale Simulation of Black Oil in a Naturally Fractured Reservoir Through Discrete Fracture Networks and Homogenized Media. *SPE Res Eval & Eng* **11** (4): 750–758. SPE-103901-PA. https://doi.org/10.2118/103901-PA.

Liu, S., Valkó, P. P., McKetta, S. et al. 2017. Microseismic Closure Window Characterizes Hydraulic-Fracture Geometry Better. *SPE Res Eval & Eng* **20** (2): 423–445. SPE-179116-PA. https://doi.org/10.2118/179116-PA.

Lutins, E. 2017. DBSCAN: What Is It? When to Use It? How to Use It. https://medium.com/@elutins/dbscan-what-is-it-when-to-use-it-how-to-use-it-8bd506293818 (accessed 19 February 2020).

Maxwell, S.C., Mack, M., Zhang, F. et al. 2015. Differentiating Wet and Dry Microseismic Events Induced During Hydraulic Fracturing. Paper presented at the Unconventional Resources Technology Conference, San Antonio, Texas, 20—22 July. URTEC-2154344-MS. https://doi.org/10.15530/URTEC-2015-2154344.

Moinfar, A., Varavei, A., Sepehrnoori, K. et al. 2014. Development of an Efficient Embedded Discrete Fracture Model for 3D Compositional Reservoir Simulation in

Fractured Reservoirs. *SPE J.* **19** (2): 289–303. SPE-154246-PA. https://doi.org/10.2118/154246-PA.

Sun, W. 2019. *A Scenario Management Platform that Incorporates Statistic and Simulation for Unconventional Field Development*. MS thesis, The University of Texas at Austin, Austin, Texas (May 2019).

Warpinski, N. 2009. Microseismic Monitoring: Inside and Out. *J Pet Technol* **61** (11): 80–85. SPE-118537-JPT. https://doi.org/10.2118/118537-JPT.

Warpinski, N. R., Mayerhofer, M., Agarwal, K. et al. 2013. Hydraulic-Fracture Geomechanics and Microseismic-Source Mechanisms. *SPE J.* **18** (4): 766–780. SPE-158935-PA. https://doi.org/10.2118/158935-PA.

Warren, J. and Root, P. 1963. The Behavior of Naturally Fractured Reservoirs. *Society of Petroleum Engineers Journal* **3** (3): 245–255. SPE-426-PA. https://doi.org/10.2118/426-PA.

Xu, W., Calvez, J. H. and Thiercelin, M. J. 2009. Characterization of Hydraulically-Induced Fracture Network Using Treatment and Microseismic Data in a Tight-Gas Sand Formation: A Geomechanical Approach. Paper presented at the SPE Tight Gas Completions Conference, San Antonio, Texas, USA, 15–17 June. SPE-125237-MS. https://doi.org/10.2118/125237-MS.

Xu, Y. and Sepehrnoori, K. 2019. Development of an Embedded Discrete Fracture Model for Field-Scale Reservoir Simulation with Complex Corner-Point Grids. *SPE J.* **24** (4): 1552–1575. SPE-195572-PA. https://doi.org/10.2118/195572-PA.

Xu, Y., Filho, J. S., Yu, W. et al. 2017. Discrete-Fracture Modeling of Complex Hydraulic-Fracture Geometries in Reservoir Simulators. *SPE Res Eval & Eng* **20** (2): 403–422. SPE-183647-PA. https://doi.org/10.2118/183647-PA.

Xu, Y., Yu, W. and Sepehrnoori, K. 2019. Modeling Dynamic Behaviors of Complex Fractures in Conventional Reservoir Simulators. *SPE Res Eval & Eng* 22 (03): 1110–1130. SPE-194498-PA. https://doi.org/10.2118/194498-PA.

Yu, X., Rutledge, J., Leaney, S. et al. 2016. Discrete-Fracture-Network Generation from Microseismic Data by Use of Moment-Tensor- and Event-Location-Constrained Hough Transforms. *SPE J.* **21** (1): 221–232. SPE-168582-PA. https://doi.org/10.2118/168582-PA.

Vita

Mohammed AlQassab is from Qatif city in the Eastern Province of Saudi Arabia. He graduated from Missouri University of Science and Technology in 2013 with a bachelor's degree in petroleum engineering. Then, he worked for Saudi Aramco for five years. He has experience in reservoir management and production engineering. In 2018, he was sponsored by Saudi Aramco to pursue his master's degree at The University of Texas at Austin. He will resume working for Saudi Aramco after completing the graduate program.

Email: mzag2d@gmail.com

This thesis was typed by the author.